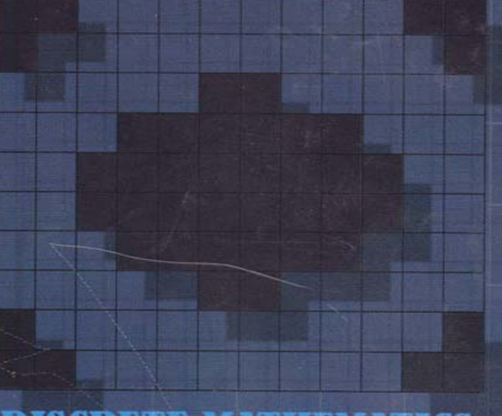
الرياضيات المتقطعة في علم الحاسوب

لطلبة علم الحاسوب وهندسة البرمجيات



DISCRETE MATHEMATICS IN COMPUTER SCIENCE

الدكتور

ناصر حسين سلمان المفرجي

مركز بحوث الفضاء - بغداد سابقاً
جامعة الزرقاء الاهلية حاليا
كلية العلوم وتكنولوجيا المعلومات
قسم الحاسوب

الدكتورة

جلنار محمد هادي الجبوري

جامعة المستنصرية - كلية العلوم
قسم الفيزياء



2008

الرياضيات المتقطعة في علم الحاسوب

DISCRETE MATHEMATICS

IN

COMPUTER SCIENCE

لطلبة علم الحاسوب وهندسة البرمجيات

FIRST EDITION

تأليف

الدكتور ناصر حسين سلمان المفرجي
مركز بحوث الفضاء - بغداد سابقاً
جامعة الزرقاء الاهلية حالياً
كلية العلوم وتكنولوجيا المعلومات
قسم الحاسوب

الدكتورة جلنار محمد هادي الجبوري
الجامعة المستنصرية - كلية العلوم
قسم الفيزياء

دار وائل للنشر

الطبعة الأولى

2008

رقم الايداع لدى دائرة المكتبة الوطنية : (2007/7/2256)
المفرجي، ناصر حسين
الرياضيات المتقطعة في علم الحاسوب / ناصر حسين سلمان المفرجي، جلنار محمد هادي الجبوري. - عـمان ، دار وائل ، 2007 .
(333) ص
ر.إ. : (2007/7/2256)
الواصفات: الرياضيات / الحواسيب / علم الحاسوب / البرمجة / المقررات الدراسية
* تم إعداد بيانات الفهرسة والتصنيف الأولية من قبل دائرة المكتبة الوطنية

****

رقم التصنيف العشري / ديوي : 510.05285
(ردمك) ISBN 978-9957-11-719-1

* الرياضيات المتقطعة في علم الحاسوب
* الدكتور ناصر المفرجي – الدكتورة جلنار الجبوري
* الطبعــة الأولى 2008



دار وائــل للنشر والتوزيع
* الأردن – عمان – شارع الجمعية العلمية الملكية – مبنى الجامعة الاردنية الاستثماري رقم (2) الطابق الثاني
هـاتف : 00962-6-5338410 – فاكس : 00962-6-5331661 - ص. ب (1615 - الجبيهة)
* الأردن – عمـان – وسـط البـلد – مجمـع الفحيص التجــاري- هــاتف: 00962-6-4627627
www.darwael.com
E-Mail: Wael@Darwael.Com



LIST OF SYMBOLS قائمة الرموز المستخدمة

1- الرموز المستخدمة في علم منطق الحاسوب Computer Logic Notations

- الاداة $\wedge$ وتعني *AND Logic*
- الاداة $\vee$ وتعني *Inclusive OR Logic*
- الاداة $\oplus$ وتعني *Exclusive OR Logic* وهي اما هذه الجملة او تلك لكن ليس كلاهما *either or not both*.
- الاداة $\neg$ وتعني **not** وهي تعني نفي الجملة التي تسبقها.

$(p \wedge q)$: *p and q*, $\wedge$ ترمز الى الاداة المنطقية *AND* بين الجملتين

$(p \vee q)$: *p or q* , $\vee$ ترمز الى الاداة المنطقية *Inclusive OR* بين الجملتين

$p \oplus q$: *p or q , but not both* $\oplus$ ترمز الى الاداة المنطقية *Exclusive OR* بين الجملتين

$\bar{p}$ او $(\neg p)$: *not p* $\neg$ تعني نفي الجملة

p implies q $(p \Rightarrow q)$, $(p \rightarrow q)$ ترمز الى ان p ضمنياً تشتمل على

$p \leftrightarrow q$ *biconditional* $\leftrightarrow$ *if and only if*   ترمز الى اداة الشرط الثنائي

$P \equiv Q$ *or* $P \Leftrightarrow Q$: *p and q are logically equivalent*   التكافؤ المنطقي

$\forall$ for all   ترمز الى الكل

$\exists$ there exist   ترمز الى انه يوجد

$\forall x P(x)$: *for every x, P(x)*   لكل القيم فان الجملة صحيحة

$\exists x P(x)$: *for some x, P(x)*   لبعض القيم فان الجملة صحيحة

Set notations **رموز المجموعة**

$\{x_1,....,x_n\}$: set consisting of the elements $x_1,....,x_n$ تمثل مجموعة من العناصر

$\{x \mid p(x)\}$: *set consisting of those elements x satisfying property* $p(x)$

تمثل مجموعة تشتمل على العناصر x التي تحقق الخاصية $p(x)$

$x \in A$: *x is an element of A*

$x \notin A$: *x is not an element of A*

$A = B$:*set quality* (*A and B have the same elements*) ترمز الى تساوي المجموعات

$|A|$:*number of element in A* يمثل عدد العناصر في المجموعة

ϕ :*empty set* ترمز الى المجموعة الخالية

:*A is a subset of B* ترمز الى المجموعة الجزئية

$P(A)$:*power set of A(all subset of A)* كل المجموعات الجزئية

$A \bigcup B$:*A union B(all elements in A or B)* تمثل اتحاد المجموعات

$A \bigcap B$:*A intersect B* (*all elements in A and B*) تمثل تقاطع المجموعات

$\bigcup_{i=1}^{n} X_i$:*union of* $X_1,....,X_n$ *(all elements that belong to at least one of* $X_1,.X_2....,X_n$)

$\bigcup_{i=1}^{\infty} X_i$:*union of* $X_1,.X_2,....$ (*all elements that belong to at least one of* $X_1,.X_2.,$)

$\bigcap_{i=1}^{n} X_i$:*intersection of* $X_1,....,X_n$ (*all elements that belong to every one of* $X_1,.X_2....,X_n$)

$\bigcap_{i=1}^{\infty} X_i$: intersection of $X_1,.X_2,....$ (all elements that belong to every one of $X_1,.X_2,....$)

$\bigcup S$: *union of S (all elements that belong to at least one set in S)*

اتحاد عائلة من المجموعات

$\bigcap S$: *intersection of S (all elements that belong to every set in S)*

تقاطع عائلة من المجموعات

$A - B$:*set difference(all elements in A but not in B)* تمثل فرق المجموعات

$\overline{A}$: *complement of A(all elements not in B)* تمثل المجموعة المتممة

(x, y) : *ordered pair* الزوج الُمرَتب

$(x_1,...., x_n)$ *n-tuple* المتعدد المرتب

AxB :*Cartesian product of A and B[pairs(x,y) with x in A and y in B]*

الضرب الكارتيزي لمجموعتين

$\{S_n\}$ **is entire sequence** **يرمز لمتتالية كاملة** $S_1, \quad S_2 ,...... \quad S_n ,......$

$\sum_{i=m}^{n} a_i$) *sum* (*or sigma*) *notation* تسمى رموز الجمع

$\prod_{i=m}^{n} a_i$ (*product notation* تسمى رموز الضرب

$\sum_{i=0}^{n} ar^i$ **is geometric sum** $a + ar + ar^2 + ... + ar^n$ **يمثل المجموع الهندسي**

X^* (*Strings*على **X** ومن ضمنها الـ) *Null string*) وترمز الى مجموعة كل الـ

X^+ **X**على *null strings* وترمز الى مجموعة كل الـ

$\sum_{i=1}^{n}(ai + b)$

Arithmetic series وترمز الى المتسلسلة الحسابية

$R(xRy)$ تمثل علاقة حيث x ترتبط بـ y

$f : D \to R$ دالة من D الى R

$f(x) = x + 2 \quad or \quad y = x + 2$ تمثل دالة ذات متغير واحد

$f(x,y) = x\,y + x^2$ تمثل دالة ذات متغيرين

$R_2 \circ R_1 = \{(x,y) \mid (x,y) \in R_1 \;\; and \;\; (x,y) \in R_2 \; for \;\; some \;\; y \in Y$

تمثل علاقة مركبة

$f \circ g(x)$ ("*f* ***composed with*** **g**") , ***f*** **circle g"** تمثل الدالة المركبة

تمثل معكوس الدالة $f^{-1} = \{(y,x) \mid (x,y) \in f\}$

الفهرس

الموضوع	الصفحة

الفصل الاول

الرياضيات المتقطعة لعلم الحاسوب

DISCRETE MATHEMATICS FOR COMPUTER SCIEINCE

الفصل الثاني

الاستنتاج (الاستقراء)الرياضي

MATHEMATICAL INDUCTION

الموضوع	الصفحة

الفصل الثالث

المجموعات والعلاقات والدوال

Sets, Relations and functions and Notations

الموضوع	الصفحة

الفصل الرابع

انظمة العــد

Numbering systems

الموضوع	الصفحة

الفصل الخامس

الجبر البولياني

Boolean Algebra

الفصل السادس

الخوارزميات

Algorithms

الموضوع	الصفحة

الموضوع	الصفحة

الفصل الثامن

الشجرة Tree

المقدمة

بعد الاتكال على الله وفقنا الى تأليف هذا الكتاب باللغة العربية ليكون مناراً مضافاً الى سلسلة الكتب العربية القليلة جداً في هذا المجال. ان الرياضيات المتقطعة مادة مقررة ومادة تخصص في علم الحاسوب , لذا فهي مهمة واساسية في منهاج طلبة علم الحاسوب في الجامعات. وبعد الخبرة التدريسية الكبيرة في قسم الحاسوب , أخذتُ بنظر الاعتبار عرض مواضيع هذا الكتاب بشيء من التفصيل والوضوح مدعوماً بامثلة توضيحية عديدة وباسلوب علمي متسلسل ومفهوم. بالاضافة الى ذلك وضعتُ معنى المصطلح بما يقابله باللغة الانكليزية للمحافظة على معناه. واشتمل الكتاب ايضا على قائمة بالرموز المستخدمة في جميع فصول هذا الكتاب .

وتاتي اهمية تاليف هذا الكتاب من ان الرياضيات المتقطعة تدخل بشكل كبير في علم الحاسوب , فهي تدخل في تحديد القيمة الحقيقية للعبارات والجمل في علم الحاسوب سواء كانت شرطية ام غيرها وتدخل في علم منطق الحاسوب وتنظيمه(المنطق وتصميم الدوائر الالكترونية الرقمية, والتكافؤ المنطقي), وفي البرهان والاستقراء الرياضي ونظرية المخططات وشجرة المعلومات والمجموعات وعملياتها, طرق العد الاساسية, انظمة العد والجبر البولياني , الخوارزميات والعلاقات والدوال, وتمثيل البيانات, الكميات المحددة وغير المحددة , المعادلات التكرارية والمعادلات الذاتية العكسية وغيرها .وبذلك فأنَّ الرياضيات المتقطعة تزود المؤسسات المتخصصة بالرياضات (*Mathematical Foundations*) بعدة مسا قات لعلم الحاسوب, اهمها تشمل تراكيب البيانات (*Data Structure*) والخوارزميات (*Algorithms*) ومترجمات الحاسوب (*Compilers*) وانظمة التشغيل (*Operating Systems*) .

اشتمل الكتاب على 8 فصول مدعومة بالامثلة والاشكال التوضيحية , حيث اشتمل الكتاب على اكثر من 200 مثالاً توضيحياً و 30 جدولاً و 60 شكلاً توضيحياً .

تناول الفصل الاول الجمل الرمزية والبسيطة والمركبة , العبارات الشرطية والتكافؤ المنطقي بانواعها وقيمها, والمنطق وقوانينه, وانواع البرهان وغيرها .

وتناول الفصل الثاني الاستقراء الرياضي او الاستنتاج الرياضي كواحد من الطرق العامة في البرهان .

وتناول الفصل الثالث بالتفصيل المجموعات ورموزها وعملياتها , والدوال والعلاقات المختلفة .

وتناول الفصل الرابع انظمة العد وعلاقاتها مع بعضها .

وتناول الفصل الخامس الجبر البولياني والدوائر الالكترونية لأختبار الجمل و بناء الدوائر الاكترونية المنطقية.

اما الفصل السادس فتناول الخوارزميات .

وتناول الفصل السابع نظرية المخططات .

واخيراً الفصل الثامن تناول شجرة المعلومات .

وقد تم الاعتماد على احدث المصادر في تأليف هذا الكتاب وخاصة المصدر الاول (انظر المصادر في نهاية الكتاب)

واخيراً اقدم كتابي هذا للقارئ العربي وللمكتبة العربية خدمة للانسان العربي وبملاحظاتكم عن مواضيع هذا الكتاب سيكون الكتاب القادم باذن الله في طبعة اكثر فائدة

والله ولي التوفيق لنا ولكم

المؤلفان

هجري :ربيع الثاني 1428 هـ

ميلادي حزيران 2007

عمّان –الاردن

قائمة الرموز المستخدمة

LIST OF SYMBOLS

الفصل الاول

الرياضيات المتقطعة لعلم الحاسوب

DISCRETE MATHEMATICS FOR COMPUTER SCIEINCE

1-1 المقدمة :

اكتسبت الرياضيات المتقطعة شعبية واسعة خلال العقود الأخيرة بسبب تطبيقاتها الواسعة في علوم الحاسوب, **حيث تشتمل الرياضيات المتقطعة لعلم الحاسوب دراسة الفروع التالية:**

1- طرق الرياضيات المتقطعة للتطبيقات في علم الحاسوب (*Discrete Mathematical Methods for Applications in Computer Science*)

2- الإستنتاج او الاستقراء الرياضي (*Mathematical Induction*) وهو مايستخدم من خلال الرياضيات (*Mathematics*) وعلم الحاسوب (*Computer Science*)على السواء , خاصة ان الإستنتاج الرياضي مفيد في الرياضيات المتقطعة والذي يعتبر واحدا من بعض الطرق العامة في البرهان (***Proof***) وكما سنرى ذلك في الفصل الثاني.

3- نظرية المخططات *Graphs theory*

4- شجرة المعلومات (*Trees*)

5- المجموعات(*Sets*)

6- العلاقات المتكافئة (*Equivalence Relations*)

7- الدوال (*Functions*)

8- المجموعات المرتبة الجزئية (*Partially Ordered* Sets)

9- المرتبة الاعلىBig-O

10- Big-Omega

11- المعادلات التكرارية والمعادلات العكسية (*Recursion and Recurrence Equations*)

12- الكميات المحددة وغير المحددة(*Finite and Infinite sums*)

13- طرق العد الاساسية (*Basic Counting Methods*)

14- الاحتمالية وسلسلات ماركوف (*Probability and Markov Chains*)

15- الجبر الخطي وتطبيقاته في علم الحاسوب (*Linear Algebra and its Application in Computer Science*)

2-1 علم منطق الحاسوب وتنظيمه

COMPUTER LOGIC AND ORGANIZATION

يشتمل علم منطق الحاسوب وتنظيمه على المفاهيم التالية :

1- أنظمة الرقم الثنائي (*Binary Number Systems*)

2- تمثيل البيانات (*Information Representation*)

3- الجبر البولياني (*Boolean Algebra*)

4- تصميم الدوائر الرقمية والصمامات (*Gates and Digital Circuit Design*)

5- تصميم حاسوب مبسط (*Designing a Simple Computer*)

6- محاكاة البرمجيات (*Software Simulation*)

ان الرياضيات المتقطعة تزود المؤسسات المتخصصة بالرياضات (*Mathematical Foundations*) بعدة مسا قات لعلم الحاسوب تشمل تراكيب البيانات (*Data Structure*) والخوارزميات (*Algorithms*) ومترجمات الحاسوب (*Compilers*) ونظرية الاتمتة (*Automata Theory*) ,واللغات التي تستخدم لنمذجة اللغات الطبيعية ولاتصال (communicate) الحواسيب بعضها بالبعض الاخر (*Formal Languages*) وانظمة التشغيل *Operating* (*Systems*) .

اما علم الرياضيات الاعتيادي (*Mathematics*) فانه عرف للتعامل مع مجموعة القيم المعدودة او المحدودة وليست القيم المستمرة (*not Continuous*) .

3-1 فكرة الجملة في علم الحاسوب The Idea of the Statement

قبل ان نبين فكرة الجملة (*Statement*) في علم الحاسوب لابد من ذكر تعاريف بعض المصطلحات المهمة في الرياضات المتقطعة وهي:

1- الرياضات المتقطعة (*Discrete Mathematic*) هي الرياضات التي نستخدمها لتحليل العمليات المتقطعة (*Discrete Processes*) .

2- العملية المتقطعة (*Discrete Process*): هـي العمليـة التـي يـتم تنفيـذها خطـوة بخطـوة (*step-by-step*) ومثال عليها حاصل ضرب رقم مؤلف من ثلاثة ارقام برقم اخر مؤلف من رقمين مثلا , فاننا نبدأ بضرب الأحاد من الرقم الثاني في الأحاد من الرقم الاول ثم العشرات ,ثم المئات من الرقم الاول ثم نعـود في الخطـوة الثانيـة لضرب رقم العشرات من الرقم الثاني بكل ارقام العدد الاول بنفس الاسلوب السابق مع ترتيب النتيجـة تحـت مرتبة المئات من الرقم الاول لنبدأ الخطوة الثالثة وهـي جمـع النتـائج بالطريقـة الاعتياديـة لعمليـات جمـع الارقام. ومثال اخر عن العملية المتقطعة هو طريقة حل المعادلات الرياضية (*Solving an Equations*) .

3- الرياضيات غـير المتقطعـة (*Continuous Discrete*) : تصـور كـرة متحركـة في الهـواء فـان الرياضـيات التـي تستخدم لتحليل عملية حركة هذه الكرة مـن ناحيـة حسـاب سرعتهـا , ارتفاعهـا , الـزمن الـذي تسـتغرقه في الحركة لغاية سقوطها ....الخ, تسمى رياضيات غير متقطعة ومثال عليها الــ(*The Calculus*).

4- الخوارزميات (*Algorithms*) : هي سلسلة من الاوامر (*Instructions*) التـي تنفـذ خطـوة بخطـوة لتنفيـذ عملية ما (*Process*). ولها الخصائص التالية:

1- الدقة ***precision*** : ان خطوات الخوارزمية توضع او ينصُّ عنها بدقة لكي يمكن كتابة الخوارزميـة في لغات البرمجة وتنفيذها بواسطة الحاسوب.

2- خاصية الحصول على نتيجة واحدة بكل خطوة بعد تنفيذها ***uniqueness***:حيث النتـائج الوسـطية (*intermediate results*) لكل خطوة بعد تنفيذها تعرَّف بنتيجـة وحيـدة, اي كـل خطـوة وسـطية للخوارزمية تنتج نتيجة واحدة فقط وتعتمد تلك النتيجة على المدخلات وعـلى نتـائج الخطـوات التـي تسبقها فقط .

3- خاصية المحدود ***Finiteness*** : وهي ان الخوارزمية تتوقف بعد عدة اوامر محددة قد تم تنفيذها.

4- المدخلات ***inputs*** : الخوارزمية تستلم مدخلات اي ندخل لها قيم .

5- المخرجات ***outputs*** :الخوارزمية تنتج مخرجات اي نحصل على النتيجة المطلوبة من الخوارزمية.

6- العمومية ***generality*** : الخوارزمية تطبَّق الى مجموعة من المدخلات ولم تخص بيانات دون اخرى.

5- علم الحاسوب (***Computer Science***) : ان علم الحاسوب هو علم معالجة المعلومـات او العلـم الـذي يعنـى بتمثيل ومعالجة البيانات.

"Computer science is the discipline concerned with representation and processing of information."

وهو ايضاً دراسة الحواسيب ودراسة كيف يمكن كتابة برامج الحاسوب , و هو الدراسة المتعلقة بالاستخدامات والتطبيقات للحاسوب وبرمجياته , حيث ان الحاسوب هو المسؤول عن بناء وتصميم البرمجيات لذا فانه يمكن القول بان علم الحاسوب يختص بدراسة الخوارزميات (خواصها التركيبية , المكونات المادية اللازمة لتنفيذها المتعلقة بتصميم الحاسوب ولغاته واخيرا تطبيقات الخوارزميات وتشمل تصميم الشبكة العنكبوتية , نماذج التصاميم وعلم المعلومات (*network design, ocean modeling, bioinformatics*).

4-1 العبارة اللغوية The Sentence

تُقسَّم العبارة اللغوية الى الاقسام التالية :

1- جملة لغوية معلنة (*Declarative Sentence*) وهي ثلاثة انواع , اما عبارة صحيحة *True* او عبارة خاطئة *False* او عبارة مبهمة (*Ambiguous*) .

2- اسئلة (*Questions*)

3- جملة نداء او تعجب (*Exclamations*)

5-1 الجملة Statement

ان العبارة (*proposition*) تُعتبر جملة (*statement*) عندما تصبح اما صحيحة او خاطئة, حيث هذا مفهوم الجملة في علم الحاسوب .وبشكل عام فان الـ (*statement*) هي جملة معلنة (*Declarative Sentence*) , وعندما تكون كذلك , فانه يمكن تصنيفها الى جملة صحيحة *True* او جملة خاطئة *False* او مبهمة (*Ambiguous*) .لكن ليس الى صحيحة او خاطئة في نفس الوقت او الى جميعها (صحيحة , خاطئة , مبهمة) في نفس الوقت . هذه الصفة في تصنيف الـ (*Statements*) جعلها تختلف عما هو للاسئلة والتي يمكن ان تسأل او للاوامرالتي يمكن اعطائها اوللنداء بشكل منادادة, لكن الجملة في علم الحاسوب يمكن تصنيفها فقط الى صحيحة او خاطئة لاغير.

أمثلة على الجمل Statements

مثال 1-1

1- بغداد عاصمة العراق.

2- 2 هو رقم زوجي واقل من 20 .

3- إذا كان 2 هو عدد زوجي فان (2+2) هو ايضا عدد زوجي .

4- العدد خمسة مضاف اليه العدد سبعة هو العدد اثنى عشر.

وهذه كلها جمل لغوية أما الجمل الرياضية او الحسابية فهي مانستخدمه في البرمجة مثل : "اذا كانت x اكبر من 3 , فان y تساوي 3 ", اي:

$$\text{"If } x > 3 \text{ then } y = 3\text{"}$$

بمعنى انه اذا تحققت الجملة" $x > 3$" اي أصبحت *True* فان الجملة الثانية " $y = 3$ " تتحقق وعكس ذلك لن تتحقق الجملة الثانية . وهذا يعني ان قيمة x يجب ان تكون معرفة في البرنامج قبل موضع الجملة الشرطية اعلاه فيه , والاّ لن يتم فحص الشرط.

مثال 1-2 : اي من العبارات (*propositions*) التالية صحيحة او خاطئة وليس كلاهما:

1- الأعداد الصحيحة الموجبة فقط والتي تقسم الرقم 7 هي **واحد** والرقم 7 نفسه . وهذه تمثل جملة صحيحة وبناء على ذلك يدعى الرقم الصحيح ***n*** رقما أولياً اذا كان $n > 1$ وتكون الاعداد الموجبة الصحيحة فقط والتي تقسم ***n*** هي **1** و ***n*** نفسه . وهنا نستطيع ان نقول الجملة السابقة بطريقة اخرى وهي ان الرقم 7 هو عدد اولي.

2- لكل عدد صحيح موجب ***n*** , يوجد عدد أولي أكبر من ***n*** . وهذه جملة صحيحة ويمكن ان نقولها بشكل اخر : هناك عدد غير محدد من الاعداد الاولية .

3- إن الارض هي الكوكب الوحيد في الكون وعليه الحياة . وهي إما جملة صحيحة او إنها جملة خاطئة حيث لحد الان لم يعرف أحداً الاجابة على هذه الجملة .

1-6 الجمل الرمزية والبسيطة والمركبة

Symbolic , Simple and Compound Statements

لتمثيل الجُمل الرمزية فاننا نسخدم حروف اللغة الانكليزية الصغيرة (*Small Letters*) وعادة ماتكون $(\ p,q,r,or\ s\ )$, ونستخدم حروف انكليزية مختلفة عادة ماتكون $(\ w,x,y,or\ z\ )$ لتقوم مقام المتغيرات (*Variables*) التي حولها بنيت الجملة المعنية, فمثلا لو اردنا ان نقول ان الجملة التالية تقوم حول المتغير ***x*** فاننا نستخدم $p(x)$ وليس ***p*** فقط للدلالة على المتغير ***x*** , وفي هذه الحالة فاننا نقرأ $p(x)$ كمايلي : $(p\ \ of\ \ x)$ أو $(p\ \ about\ \ x)$. وهنا لو قلنا ان الرمز ***p*** تقوم مقام الجملة $(x < 10)$ او ان $p(x)$ تمثل الجملة $(x < 10)$ فان ***p*** و $p(x)$ هما الجمل الرمزية لاحتوائها على رموز وان ***x*** هي المتغير (*Variable*) .

كما انه احيانا يتم تمثيل الجمل الرمزية بمايلي : مثلا $(\ p : 1+1=3\ )$ تعني ان ***p*** رمز لتعريف الجملة: $(1+1=3)$, وهي جملة خاطئة طبعاً. ويمكن ان نسمي الجمل

الرمزية صيغ منطقية (*Logical Formula*) ونشير الى $(\neg p)$ و **p** بالجمل الرمزية لاحتوائها على رموز تقوم مقام الجمل المسماة الجمل الرمزية .

اما الجمل البسيطة (*Simple Statements*) فهي تتكون من رمز واحد , وهي اما ان تكون صحيحة *True* او تكون خاطئة (*False*) وهي تدل على مفهوم واحد . لكن الجمل المركبة (*Compopound Statements*) فهي تتركب من اكثر من جملة بسيطة , مثلا (**p, q**), ويكون جدول القيم الحقيقية (*Truth Table*) لها معتمدة على قيم الجملة **p** والجملة **q** وعلى الاداوات المنطقية (*Connective Symbols*) التي تربط بينهما , وهذه الادوات المنطقبة هي:

1- الاداة $\wedge$ وتعني *AND Logic*

2- الاداة $\vee$ وتعني *Inclusive OR Logic*

3- الاداة $\oplus$ وتعني *Exclusive OR Logic* وهي اما هذه الجملة او تلك لكن ليس كلاهما *either or not both*.

4- الاداة $\neg$ وتعني **not** وهي تعني نفي الجملة التي تسبقها.

ان الادوات المذكورة اعلاه تستخدم لبناء الجمل المركبة من اكثر من جملة بسيطة , مثلا الجملتين البسيطتين التاليتين " السماء تمطر , " انا استخدم مظلة المطر " يمكن ربطهما بجملة واحدة هي " السماء تمطر **و** انا استخدم مظلة المطر" وتسمى الجمل الرمزية التي تحوي ادوات الربط المنطقية (*Connective Symbols*) بالجمل المركبة الرمزية كما في الجملة التالية :

$$p \wedge (q \oplus r)$$

اما اذا اردنا ان نعبر عن الجملة $p \wedge (q \oplus r)$ برمز واحد فاننا نكتب :

" $let \quad s = p \wedge (q \oplus r)$ " حيث ان الرمز **s** يقوم مقام تلك الجملة .

وبناءاً على ماذكرناه فان اقتران (*Conjunction*) الجملة **p** والجملة **q** تمثله بالجملة المركبة $(p \wedge q)$

وعدم اقترانهما (*Discon junction*) نعبر عنه بالجملة المركبة $(p \vee q)$.

مثال 1-3

لو كانت لدينا الجُمل التالية :

p:1+1=3 , **q** :A week is 7 days

فان العبارة $(p \wedge q)$ هي :

$$(p \wedge q) : 1+1 = 3 \text{ and a week is 7 days}$$

والعبارة $(p \vee q)$ هي :

$(p \vee q): 1+1=3$ OR a week is 7 days ▲

ونود ان نُبيِّن هنا ان نفي الـ AND كما في $\neg(p \wedge q)$ فانه يعني $(\neg p \vee \neg q)$, وان نفي الـ OR كما في $\neg(p \vee q)$ فانه يعني $(\neg p \wedge \neg q)$, وهذه الحقائق تشكل قوانين دي موركان كما سنأتي عليها لاحقاً.

ولوصف القيم الحقيقية للعبارة $(p \wedge q)$ والعبارة $(p \vee q)$ فاننا نستخدم جدول القيم الحقيقية (*Truth Table*), حيث ان قيم جدول القيم الحقيقية للجملة *p* يمثل قيم الجمل الانفرادية لها $p_1, p_2 \ldots\ldots p_n$ وهذه القيم تشمل كل الاحتمالات الممكنة لربط القيم الحقيقية للجمل $p_1, p_2 \ldots\ldots p_n$ في الجدول . ويستخدم الحرف T او الرقم واحد (1) في الجدول للدلالة على الجملة الصحيحة True , ويستخدم الحرف F او الصفر (0) للدلالة على الجملة الخاطئة False كما في الجدول (1-1) التالي:

p	*q*	*p*	$\wedge$	*q*
1	1		1	
1	0		0	
0	1		0	
0	0		0	

p	*q*	*p*	$\wedge$	*q*
T	T		T	
T	F		F	
F	T		F	
F	F		F	

p	*q*	*p*	$\vee$	*q*
1	1		1	
1	0		1	
0	1		1	
0	0		0	

(p or q)

p	*q*	*p*	$\oplus$	*q*
1	1		0	
1	0		1	
0	1		1	
0	0		0	

$(p \quad exor \quad q)$

جدول (1-1)

وهنا الكلمة **AND** تم تمثيلها بالاداة $\wedge$ والكلمة **OR** تم تمثيلها بـ $\vee$, والعبارة $(p \;\; and \;\; q)$ تُكتب بـ $(p \wedge q)$ والعبارة $(p \; or \; q)$ تكتب بـ $(p \vee q)$, أو $(\oplus \;\; either \;\; or \;\; not \;\; both)$ $(p \oplus q)$.

ويُلاحظ ايضا من الجداول اعلاه ان قيم العبارة $(p \wedge q)$ تكون True فقط عندما تكون كل من ***p*** و ***q*** صحيحة وعكس ذلك تكون False . اما بالنسبة للعبارة $(p \vee q)$ فانها اعتبرت صحيحة اذا كان كل من **p** او ***q*** او كلاهما *True* وان $(p \vee q)$ هي خاطئة فقط عندما كل من ***p*** و ***q*** كان خاطئا, لكن العبارة $(p \;\; exor \;\; q)$ فانها تكون صحيحة (T) اذا كان كل من ***p*** او ***q*** صحيحا لكن ليست عندما كلاهما صحيح.

اما الجملة $not"x < 2"$ فهي تفسر (x ليست اقل من 2)وهنا سنستخدم الاداة $\neg$ لتقوم مقام *not* . ولو كانت *p* تقوم مقام (x ليست اقل من 2) فان $\neg p$ تقوم مقام الجملة (ليست الحالة ان $"x < 2"$ او (x ليست اقل من 2).

مثال 1-4

اذا كانت $(p(x) : x > 0)$ هي جملة بسيطة , و $(q(x) : x < 10)$ هي جملة بسيطة اخرى فان الجملة $(x > 0 \;\; and \;\; x < 10)$ هي جملة مركبة , حيث اننا لانستخدم التعبير الرياضي الجبري $(0 < x > 10)$ في برمجيات الحاسوب بل انها تكتب $(x > 0 \;\; and \;\; x < 10)$ اي $(p \wedge q)$ لان هذا النوع من الجمل فقط يستطيع الحاسوب ان يحدد فيما لو كانت جملة صحيحة او جملة خاطئة .

▲

وهناك حالات اخرى تربط بين الجُمل ***p*** او ***q*** هي ان ***p*** تدل ضمنيا على ***q*** اي *(p implies q)* ويرمز لها بـ $(p \Rightarrow q)$ وتكون هذه العبارة خاطئة عندما الجملة ***p*** صحيح *(True)* والجملة ***q*** خاطئة , ونفس الشئ ينطبق على جميع العبارات التالية لانها متكافئة *(Equivalent Statements)* :

(if p then q), (p only if q), (q if p)

انظر القيم الحقيقية (*truth tables*) في الجدول (1-2) التالي:

p	q	p	$\rightarrow$	q
T	T		T	
T	F		F	
F	T		T	
F	F		T	

If p then q

p	q	p	$\Rightarrow$	q
T	T		T	
T	F		F	
F	T		T	
F	F		T	

(*p implies q*)

جدول (1-2)

اما اذا كان الرمز $\leftrightarrow$ الذي يعني " اذا وفقط اذا " "***if and only if*** " (يعني وجود شرطين) يربط بين جملتين فان قيمتة ستكون صحيحة *True* فقط عندما ***p*** **و** ***q*** اما كلاهما *True* او كلاهما *False* كما في الجدول (1-3) ادناه:

p	q	p	$\leftrightarrow$	q
1	1		1	
1	0		0	
0	1		0	
0	0		1	

if and only if

جدول (1-3)

اخيراً بقي لدينا نفي الجملة ***p*** وهو ***not p*** ويرمز لذلك بـ $\bar{p}$ او $(\neg p)$ وجدول القيم *"Truth table"* لها جدول (1-4) كمايلي:

p	p ¬
1	0
0	1

جدول (1-4)

فمثلا لو كانت $p: x \geq 1$ فان $\bar{p}$ او $(\neg p)$ هي $x < 1$.

ونلاحظ من الجداول التي اشرنا اليها سابقاً ان عدد القيم فيه تساوي احتماليات الجمل المكونّة له اي عندما تكون لدينا جملتين في الجدول كما هو الحال في $(p \wedge q)$ و في $(p \vee q)$ فان القيم تكون اربعة فقط $(probability = 2^2)$ لان كل جملة لها قيمتان *True* وقيمتان *False* يقابلها نفس العدد من القيم للجملة الثانية وبالتالي تكون نتيجة الجمل المركبة $(p \wedge q)$ و $(p \vee q)$ اربعة قيم ايضا . وعندما تكون لدينا جملة واحدة كما في $(\neg p)$ فانه لدينا قيمتان فقط اما *True* او *False* لاغيرذلك . لاحظ الجداول المذكورة اعلاه.وسنرى عندما يحوي الجدول ثلاثة جمل فان عدد القيم فيه سيكون ثمانية لكل جملة على انفراد ولحاصل نوع الربط بينهما كما هو في الجدول (1-5) اناه .

p	*q*	*r*
T	**T**	**T**
T	**T**	F
T	F	T
T	F	F
F	T	T
F	T	F
F	F	T
F	F	F

جدول (1-5)

1-7 تحديد جداول القيم لجمل رمزية اخرى More about Determining the truth of symbolic statements

عرفنا ان جداول القيم الحقيقية (*Truth Tables*) هي الجداول التي تبين لنا كيف نحدد صواب وخطأ العبارات المركبة باستخدام اداة ربط واحدة مستعينين بقيم الصواب والخطأ للعبارات الممثلة رمزياً في الجدول . وتكون هذه الجداول مكونة من عدة صفوف rows , يمثل الصف الاول وهو الصف رقم صفر (*zero row*) رموز العبارات التي تكوّن الجدول , اما الصفوف الاخرى فتمثل القيم الصحيحة والخاطئة للعبارات الرمزية وتكون مرتبة ترتيباً ابجدياً عكسياً (*Reverse alphabetical order*) اي تمثل العبارات اولا بالقيمة **T** ومن ثم بالقيمة **F** وسميت ترتيب عكسي لان **T** قبل **F** بالترتيب الابجدي .انظر الجدول (1-6) ادناه:

p	q	$p \oplus q$	*top row is zero row*
1	1	0	1st
1	0	1	2nd
0	1	1	3rd
0	0	0	4th

جدول (1-6)

ومن الجدول (1-6) إذا كانت p و q كلاهما F أو T فان جملة $p \oplus q$ هو F أو نعبِّر عنها بـ (0).

من السهل سابقا تحديد صواب الجملة أو خطاها , نعني p , q , $p \wedge q$ و $p \vee q$. لكن في الجمل المركبة الأخرى مثل $\neg p \vee q$, $(p \vee q) \wedge \neg (p \wedge q)$ فمن الصعب تحديد ذلك من خطوة واحدة و لذلك نستخدم الطريقة التالية وبثلاثة خطوات , تبدأ الخطوة الاولى من اليسار الاعلى انظر الجدول (1-7):-

p	q	$\neg$	p	$\vee$	q
T	T		T		T
T	F		T		F
F	T		F		T
F	F		F		F
			1		1

p	q	$\neg$	p	$\vee$	q
T	T	F	T		T
T	F	F	T		F
F	T	T	F		T
F	F	T	F		F
		2	1		1

p	q	$\neg$	p	$\vee$	q
T	T	F	T	T	T
T	F	F	T	F	F
F	T	T	F	T	T
F	F	T	F	T	F
		2	1	3	1

جدول (1-7)

مثال 5-1

بين ان العبارة ($p \oplus q$) والعبارة $(p \vee q) \wedge \neg(p \wedge q)$ بالامكان ان تبدل (*replaced*) احداهما الاخرى .

قبل البدء بحل هذا المثال لابد من توضيح مايلي:

أولاً- ان لغة الحاسوب لاتستطيع تمييز الاداة المنطقية $\oplus$. لذا من المفيد التعبير عنها وعن العبارات ($p \to q$ و $p \leftrightarrow q$) مثلا بدلالة ادوات الربط المنطقية $\neg$ and $\vee,$ $\wedge,$.

ثانياً- لماذا جداول القيم (*Truth Tables*). والجواب هو بسبب بعض الغايات العامة للغات الحاسوب وبسبب بعض لغات استعلام قواعد البيانات لاتسمح لنا بالاختبار المباشر للقيم الحقيقية للعبارة $p \oplus q$. لذا اذا اردنا ان نختبر قيم العبارة $p \oplus q$ فانه يجب ترجمة العبارة:

" *p or q , but not both* " والتي هي $(p \vee q) \wedge \neg(p \wedge q)$

الى صيغ من $\neg$ and $\vee,$ $\wedge,$ (*and , or , and not*) . والفكرة المألوفة هي ابدال الكلمة but بالاداة AND . ان هذا يعطينا العبارة " *p or q and not both p ^ q* " . وبدلالة الرموز هذا يعطينا العبارة التالية :

$$(p \vee q) \wedge \neg(p \wedge q)$$

	q	p	$\vee$	q	$\wedge$	$\neg$	(p	$\wedge$	q)
T	T	T	T	T	F	F	T	T	T
T	F	T	T	F	T	T	T	F	F
F	T	F	T	T	T	T	F	F	T
F	F	F	F	F	F	T	F	F	F

p	q		$p \oplus q$
F	T		F
T	F		T
F	T		T
F	**F**		F

جدول (1-8)

ومن الجدول (1-8) في هذا المثال, نـرى ان $(p \vee q) \wedge \neg(p \wedge q)$ و $p \oplus q$ لهـما نفس النتيجة . اذن يمكـن اسـتبدال احـدهما بـالاخر .ونعبرعنهـا بــ $p \oplus q \Leftrightarrow (p \vee q) \wedge \neg(p \wedge q)$. ان الرمـز $\Leftrightarrow$ يشـير الى عمليـة التكافئ بين العبارتين الذي يربط بينهما وهو ليست اداة منطقية ويقرأ بانه مكافئ الى وليس مسـاوياً *"is equivalent to not equals"* .

▲

مثال1-6

اذا كانت p و q لهما نفس القيم *True* او *False* فيقال عنها انهما جملتان متكافئتان منطقيـاً (*logically equivalent statements*) ويعبر عنهما بالجدول (1-9) التالي:

p	q	q $\Leftrightarrow$ p
1	1	1
1	0	0
0	1	0
0	0	1

جدول (1-9)

1-8 قاعدة الاسبقية Precedence Rule

في العبارة $(\neg r) \vee q$ استخدمنا الاقواس حول الحد $(\neg r)$ وذلك من اجل الوضوح وعـدم الالتبـاس مـثلما نستخدم ذلك في الرياضيات عندما نكتب $(-1)+4$. وطبقا لقاعدة الاسـبقية فـان الجملـة $(\neg r) \vee q$ تفسرـ دائمـا بـ $(\neg r) \vee q$ وليست $\neg(r \vee q)$. وفي الجملة الرمزية $\neg(p \wedge q) \vee r$ فان $(p \wedge q)$ اول جملـة كاملـة الى يمـين الاداة $\neg$ لـذا فـان الجملـة الرمزيـة $\neg(p \wedge q) \vee r$ تفسرحسـب قاعـدة الاسـبقية كمايـلي: $[\neg(p \wedge q)] \vee r$ وهنا تاخذ الاداة $\neg$ الاسبقية على كل الادوات الاخرى .

مثال 1-7

افرض ان ***p, q, r*** تقوم مقام الجمل اللغوية التالية:

1- محمد في المدرسة

2- احمد في المتجر

3- علي يرد على الهاتف

على التوالي , اعد كتابة الجمل الرمزية ادناه حسب قواعد اللغة الانكليزية (تكتب لغوياً) .

1- $p \wedge (\neg q)$

2- $r \vee (p \wedge q)$

3- $(\neg p) \wedge (\neg q)$

الحل:

1- محمد في المدرسة واحمد ليس في المتجر

Mohammed is at school and Ahmed is not at the store.

2- علي يرد على الهاتف او كلاً من محمد في المدرسة واحمد في المتجر

Ali is on the phone or both Ali is at school and Ahmed is at store.

Note how the word both acts like parentheses.

لاحظ ان كلمة ***both*** تعمل مثل عمل الاقواس ( ).

3- محمد ليس في المدرسة و احمد ليس في المتجر

Mohammed is not at school and Ahmed is not at the store

او يمكن كتابتها (لامحمد في المدرسة ولا أحمد في المتجر)

neither *is Mohammed at school* ***nor*** *is Ahmed at the store*



1-9 العبارات الشرطية والتكافؤ المنطقي

CONDITIONAL PROPOSITIONS AND LOGICAL EQUIVALANCE

تعريف 1-1

اذا كانت p و q هي عبارات بسيطة فان العبارة المركبة التالية :

"**اذا p ثم q** " [***if p then q***] هي عبارة شرطية ونعبِر عنها بـ $p \rightarrow q$.ان العبارة p تدعى الفرضية ***hypothesis*** (***or antecedent***) التي يتم اثباتها علمياً , والعبارة q تدعى النتيجة (***or consequent***)***conclusion***.

مثال 1-8

أعد صياغة الجمل التالية حسب الصيغة الشرطية [*if p then q*] للعبارات الشرطية التالية:

1- امنة ستكون طالبة جيدة اذا هي درست بشدة

Amna will be a good student if she studies hard

الفرضية هنا هي العبارة التي تلي *if* . اذن الصيغة المكافئة للصيغة الشرطية [*if p then q*] هي:

If *Amna studies hard*, ***then*** *she will be a good student*

2- احمد ربما ياخذ رياضيات التفاضل والتكامل فقط اذا هو طالب في السنة الثانية من كلية

Ahmed may take calculus only if he has sophomore

وهنا فقط عبارة *if* هي النتيجة وهذا يعني ان [***if p then q***] هي نفسها [***p only if q***] وستكون الصيغة المتكافئة هي :

(*If Ahmed takes calculus, then he has sophomore.*)

ان الصيغة [***if p then q***] تؤكد الفرضية ***hypothesis*** بينما الصيغة [***p only if q***] **تؤكد النتيجة** (***conclusion***) .

3- عندما تغني , اذنيي تتألم. (*when you sing , my ears hurt*).

ان كلمة ***when*** هنا تعني نفس معنى ***if*** . اذن ستكون الصيغة المكافئة هي:

if you sing ,then my ears hurt. ▲

تعريف 1-2

جدول(1-10) *Truth Table* يمثل القيم الحقيقية للعبارة $p \Rightarrow q$ (*p implies q*) , والعبارات الشرطية انفة الذكر كمايلي:

p	q	p	$\rightarrow$	q
T	T		T	
T	F		F	
F	T		T	
F	F		T	

If p then q

p	q	p	$\Rightarrow$	q
T	T		T	
T	F		F	
F	T		T	
F	F		T	

(*p implies q*)

جدول (1-10)

وبشكل عام عندما تكون الفرضية p صحيحة *True* فان العبارة $p \to q$ صحيحة اذا كانت العبارة q صحيحة , وتعتبر $p \to q$ خاطئة اذا p خاطئة *False*. اما اذا كانت الفرضية p خاطئة False فان النقطة الواضحة بديهياً فقط هي ان القيمة الحقيقية *Truth value* للعبارة الشرطية $p \to q$ يجب ان لاتتوقف على القيمة الحقيقية للنتيجة (*conclusion's truth value*) .بالرغم من ذلك فان العبارة الشرطية مثل اي عبارة اخرى يجب ان يكون لها قيمة حقيقية (*Truth value*) حتى لو كانت الفرضية خاطئة . ان التعريف المعياري (القياسي) يبين ان العبارة $p \to q$ تكون *True* اذا p هي خاطئة .

وبذلك فان $p \to q$ هي عبارة خاطئة فقط عندما p هي *true* .

مثال 1-9

-1 بيّن ان $x > 2$ هي تدل ضمنا على $x > 1$ اي ان "$x > 1$ implies $x > 2$".

الحل:

اذا كانت p تمثل الجملة $x > 2$ و q تمثل الجملة $x > 1$ وكانت كل من p و q جملتان حول المتغيرات فان المجموعة الحقيقية لـ $x > 2$ هي مجموعة جزئية للمجموعة الحقيقية لـ $x > 1$. لذا فان الجملة $x > 2$ تدل ضمنا على الجملة $x > 1$ ونعبّر عنها بـ "$x > 1 \to x > 2$".

$x > 1 = \{2, 3,4,5\ldots\}$ و $x > 2 = \{3, 4,5,6\ldots.\}$. ▲

-2 افرض ان : $q : 4 < 8 \; ; p : 1 > 2$.

نرى في هذا المثال ان العبارة p هي عبارة خاطئة *False* وان العبارة q عبارة صحيحة *True* لذا ستكون العبارة $p \to q$ هي عبارة صحيحة *True* لكن العبارة $q \to p$ ستكون خاطئة . انظر الجدول "*truth table*" (1-2) .

مثال 1-10

افرض ان p صحيحة *True* , q خاطئة *False* و r صحيحة *True* . اوجد القيمة الحقيقية *Truth value* لكل ممايأتي:

1- $(p \wedge q) \rightarrow r$

2- $(p \vee q) \rightarrow \bar{r}$

3- $p \wedge (q \rightarrow r)$

4- $p \rightarrow (q \rightarrow r)$

الحل:

سنبدل كل رمز p, q و r بقيمتها الحقيقية (*Truth value*) للحصول على القيمة الحقيقية لكل عبارة في هذا المثال وكمايلي:

1- $(T \wedge F) \rightarrow T = F \rightarrow T = true$

2- $(T \vee F) \rightarrow \bar{T} = T \rightarrow F = false$

3- $T \wedge (F \rightarrow T) = T \wedge T = true$

4- $T \rightarrow (F \rightarrow T) = T \rightarrow T = true$ ▲

في اللغة الاعتيادية (*ordinary language*) , الفرضية والنتيجة في العبارة الشرطية (*a conditional proposition*) هما مرتبطتان اعتياديا لكن في منطق الحاسوب لايتطلب منهن ان يشيرا الى نفس الموضوع . على سبيل المثال في منطق الحاسوب نسمح بالعبارة التالية :

اذا خمسة اقل من ثلاثة , فان الحرب العالمية الاولى وقعت عام 1914 .

If 5<3 , then the 1st world war was happened in 1914.

ان المنطق يتعلق بصيغة العبارات وبعلاقة العبارات لبعضها الاخر وليس بشأن الموضوع نفسه. في الحقيقة بما ان الفرضية 5<3 هي خاطئة فان العبارة " اذا خمسة اقل من ثلاثة , , فان الحرب العالمية الاولى وقعت عام 1914 " صحيحة .لاحظ ان عبارةً شرطية صحيحة تختلف من عبارة شرطية نتيجتها (*Conclusion*)صحيحة *True* .

مثال 1-11

ان المثال 1-9 الجزء الثاني اعلاه يبيّن ان العبارة $q \rightarrow p$ تكون صحيحة بينما العبارة $p \rightarrow q$ هي خاطئة . وهنا ندعو العبارة $p \rightarrow q$ بانها معكوس (***converse***) العبارة $q \rightarrow p$. وعلى اساس ذلك ان عبارة شرطية ممكن ان تكون صحيحة True بينما نقيضها هو عبارة خاطئة . ▲

مثال 1-12

اكتب كل عبارة شرطية ممايأتي بشكل رموز .و اكتب نقيض كل منها بشكل رموز وبشكل كلمات . ثم اوجد القيمة الحقيقية (*Truth value*) لكل عبارة شرطية ولكل معكوسها.

1- $if \quad 1<2, \quad then \quad 3<6$

2- $if \quad 1>2, \quad then \quad 3<6$

الحل:

1 -فيما يخص العبارة الاولى : نفرض ان $p:1<2, \quad q:3<6.$ وهاتان العبارتان يمكن تمثيلهما بشكل رموز بـ $p \to q$, وبما ان p و q كل منهما هو *True* فان العبارة $p \to q$ هي صحيحة . اما معكوس تلك العبارة فيمكن كتابته بشكل رموز بـ $q \to p$ واما بدلالة الكلمات فستكون : $if \quad 3<6, \quad then \quad 1<2$. وبما ان p و q كلاهما صحيحة فان المعكوس $q \to p$ هو صحيح ايضا .

2- فيما يخص العبارة الثانية : نفرض ان $p:1>2, \quad q:3<6.$, وهاتان العبارتان يمكن تمثيلهما بشكل رموز بـ $p \to q$, وبما ان p هي خاطئة *False* و q هي صحيحة *True* فان العبارة $p \to q$ هي صحيحة . اما معكوس تلك العبارة فيمكن كتابته بشكل رموز بـ $q \to p$ واما بدلالة الكلمات فستكون : $if \quad 3<6, \quad then \quad 1>2$. وبما ان ***p*** هي خاطئة False و ***q*** هي صحيحة *True* فان المعكوس $q \to p$ هو خاطئ . ▲

جملة مركبة اخرى مفيدة هي :

p if and only if q

مثل هذه الجملة تعتبر صحيحة بدقة عندما يكون لكل من ***p*** و ***q*** نفس القيم الحقيقية "*the same truth values*" (على سبيل المثال كلاهما *True* او كلاهما *false*).

تعريف 1-3

اذا كانت ***p*** **و** ***q*** هي عبارات بسيطة فان العبارة المركبة التالية :

" **p اذا و فقط اذا q** " [***p if and only if q***] هي عبارة شرطية باتجاهين **(*biconditional proposition*)** ونعبِر عنها بـ $p \leftrightarrow q$.

تعريف 1-4

ان القيمة الحقيقية للعبارة الشرطية $p \leftrightarrow q$ تعرف بجدول القيم ***Truth Table***(1-11) كمايلي: :

p	*q*	*p*	$\leftrightarrow$	*q*
T	T		T	
T	F		F	
F	T		F	
F	F		T	

جدول (1-11)

وباسلوب بديل يمكننا ان نكتب العبارة [***p if and only if q***] بمايلي:

" ***p*** ضرورية وشرط كافي لـ ***q*** " . كما ان العبارة [***p if and only if q***] تكتب احيانا بـ "***p iff q***"

مثال 1-13

العبارة التالية :" $1 < 5 \quad if \quad and \quad only \quad if \quad 2 < 8$" بالامكان كتابتها بشكل رموز (*symbolically*) بـ

$$p \leftrightarrow q$$

ولو عرَفنا *p* و *q* بالشكل التالي:

$p:1<5, \quad q:2<8$ فان العبارة $p \leftrightarrow q$ صحيحة *True* لأن *p* و *q* كلاهما *True* . وبطريقة بديلة يمكننا كتابة العبارة" $1 < 5 \quad if \quad and \quad only \quad if \quad 2 < 8$" بالجملة " الشرط الضروري والكافي لـ $1 < 5$ هو $2 < 8$.

في بعض الحالات عبارتان مركبتان مختلفتان يكون لهما نفس القيم الحقيقية (*Truth values*) . مثل هذه العبارات يطلق عليها متكافئة منطقيا (*logically equivalent*). ▲

تعريف 1-5

افرض ان العبارتان المركبتان **P** و **Q** متكونان من العبارات $p_1................,p_n$ فاننا نقول في هذه الحالة ان هما متكافئتان منطقيا (*logically equivalent*) ونكتب العلاقة بينهما كمايلي:

$$P \equiv Q$$

و تحقيق تلك العلاقة يعني ان اعطاء اية قيم حقيقية لـ $p_1................,p_n$ انه اما **P** و **Q** كلاهما صحيحا او اما **P** و **Q** كلاهما خاطئا.

مثال 1-14

استخدم جدول القيم "*truth table*" لأثبات ان $p \leftrightarrow q$ و $(p \to q) \wedge (q \to p)$ هما متكافئتان:

الحل:

ان الجملتين اعلاه يمكن كتابتهما بشكل متكافئ كما يلي:

$$p \leftrightarrow q \Leftrightarrow (p \to q) \wedge (q \to p)$$

ولأثبات ذلك فاننا نجد جدول القيم لـ $p \leftrightarrow q$ ونجد ايضاً جدول القيم لـ $(p \to q) \wedge (q \to p)$ ونقارن بين نتائجهما النهائية:

1- جدول القيم لـ $(p \to q) \wedge (q \to p)$ هو جدول (1-12) كالاتي:

p	$\to$	q	$\wedge$	$(q \to p)$		
T	T	T	T	T	T	T
T	F	F	F	F	T	T
F	T	T	F	T	F	F
F	T	F	T	F	T	F

جدول (1-12)

2- اما جدول القيم لـ $p \leftrightarrow q$ فهو جدول (1-13) كالاتي:

p	q	$p \leftrightarrow q$
T	T	T
T	F	F
F	T	F
F	F	T

جدول (1-13)

ومن الجدول (1-12) نرى ان النتيجة النهائية هي نفسها في الجدولين (T, F,F .T). ▲

ملاحظة: هناك فرق بين الجملة الشرطية "*conditional statement* " والاوامر الشرطية *"Conditional instructions* " . ففي لغات البرمجة *"programming languages* " ان الكلمات " IF and THEN " هي كلمات محجوزة للاوامر الشرطية التي تاخذ الصيغة التالية:

IF *statement* THEN *instruction sequence*

فاذا كانت الجملة صحيحة ام خاطئة (حسب تحديدنا لها) فان الحاسوب يقوم بتنفيذ التعليمات المتعلقة بذلك . والحاسوب ينفذ سلسلة الاوامر فقط واذا فقط *"if and only if* " الجملة صحيحة .لذلك فالامر الشرطي لايفسر بنفس طريقة تفسير الجملة الشرطية .

ان التعبير *"if and only if* " يعني استخدام الشرط مرتين "*The biconditional*" فمثلا لو قلنا :

"اذا النقود كثيرة فان الفائدة قليلة , واذا الفائدة قليلة فان النقود كثيرة " ونعبر عنها باللغة الانكليزية بـ

. "*If money is plentiful ,then interest rates are low, and if interest rates are low ,then money is plentiful*"

ولهذا السبب فان جملة مثل $(p \to q) \wedge (q \to p)$ يقال عنها ثنائية الشرط "*biconditional*" وعادة نرمز لها بـ $\leftrightarrow$ () . لذا بواسطة التعريف ان الجملة $p \leftrightarrow q$ تكافئ الجملة $(p \to q) \wedge (q \to p)$ ونكتبهما بالصيغة التالية : $p \leftrightarrow q \Leftrightarrow (p \to q) \wedge (q \to p)$

ومن جدول (1-13) القيم لـ $p \leftrightarrow q$ اعلاه نرى ان قيمة الجملة ثنائية الشرط $p \leftrightarrow q$ هي صحيحة true فقط عندما كل من p و q اما صحيحا واما خاطئاً. وماعدا ذلك تكون خاطئة. لذلك من المفيد جدا التعبير عن $p \leftrightarrow q$ بدلالة الرموز $\neg \quad and \quad \vee, \quad \wedge$.

مثال 1-15

اعد كتابة الجملة الرمزية التالية كجملة مكافئة بدون استخدام السهم

$$\neg r \to (s \vee (r \wedge t))$$

الحـل: ان الجملـة المعطـاة لهـا الصـيغة $p \to q$ (*if p then q*) وبهـذه الحالـة تكـون p تمثـل $\neg r$, و q تمثـل $s \vee (r \wedge t)$. اذن لكتابة الجملة المعطاة بدون سـهم فانـه يجـب ان نكتـب $p \to q$ بـدون سـهم . في هـذه الحالة ناخذ بديلها وهو $\neg p \vee q$ لان الجملتين متكافئتين :

$$p \to q \Leftrightarrow \neg p \vee q$$

في هذه الحالة نعوض p في الجملة $\neg p \vee q$ بـ $\neg r$ ونعوض q فيها بـ $s \vee (r \wedge t$ وهذا يعطينا الجملة التالية:

$$(\neg\neg r) \vee (s \vee (r \wedge t)) . \qquad \blacktriangle$$

10-1 قوانين De Morgan's Laws للمنطق: *De Morgan's Laws For Logic*

تعريف 1- 6 :: مثال 1-16

في هذا المثال سنتحقق من القانون الاول لقوانين (De Morgan;s Laws)

-1 $\overline{p \vee q} \equiv \bar{p} \wedge \bar{q}$ وهو نفسه $\neg (p \vee q) \Leftrightarrow \neg p \wedge \neg q$

-2 $\overline{p \wedge q} \equiv \bar{p} \vee \bar{q}$ وهو نفسه $\neg (p \wedge q) \Leftrightarrow \neg p \vee \neg q$

بواسطة كتابة جداول القيم الحقيقية للعبارات التالية :

$P = \overline{p \vee q}$ و $Q = \bar{p} \wedge \bar{q}$ فانه بالامكان التحقق بان اعطاء اية قيم حقيقية لـ p و q فانه اما P و Q كلاهما صحيحة او كلاهما خاطئة . انظر الجدول(1-14) ادناه:

p	q	$\overline{p \vee q}$		$\bar{p} \wedge \bar{q}$
T	T	F		F
T	F	F		F
F	T	F		F
F	F	T		T

جدول(1-14)

وبذلك فان P و Q متكافئين منطقياً.

مثال 1-17:

بين ان نفي العبارة $p \rightarrow q$ هو مكافئ منطقياً الى العبارة $p \wedge \bar{q}$ وهي نفسها $p \wedge \neg q$

الحل:

لحل هذا المثال يجب علينا ان نثبت العلاقة التالية :

$$\overline{p \rightarrow q} \equiv p \wedge \bar{q} \quad \text{او} \quad \neg(p \rightarrow q) \Leftrightarrow p \wedge \bar{q}$$

وبكتابة جدول القيم الحقيقية(1-15) للعبارات التالية :

$P = \overline{p \rightarrow q}$ و $Q = p \wedge \bar{q}$ فانه بالامكان التحقق بان اعطاء اية قيم حقيقية لـ p و q فانه اما P و Q كلاهما صحيحة او كلاهما خاطئة . انظر الجدول (1-15) التالي:

p	q	$\overline{p \to q}$		$p \wedge \bar{q}$
T	T	F		F
T	F	T		T
F	T	F		F
F	F	F		F

جدول(1-15)

وبذلك فان P و Q متكافئين منطقياً. ▲

مثال 1-18 :

برهن ان العبارة $p \leftrightarrow q$ هي مكافئة منطقيا الى ($p \to q \quad and \quad q \to p$)

الحل: ان جدول القيم (1-16) ادناه يبين ان :

$$p \leftrightarrow q \equiv (p \to q) \wedge (q \to p)$$ ▲

p	q	$p \leftrightarrow q$	$p \to q$	$q \to p$	$(p \to q) \wedge (q \to p)$
T	T	T	T	T	T
T	F	F	F	T	F
F	T	F	T	F	F
F	F	T	T	T	T

جدول (1-15)

وبناءاً على ماذكرناه سابقاً , فان المقارنة بين جدول القيم (1-2) للعبارات الشرطية او العبارات التي تشتمل ضمناً على عبارة اخرى (*implication*), وجدول القيم للمكافئات (*equivalence*)(1-16) يتبن انه اذا $p \Leftrightarrow q$ هي صحيحة

True (هـذا يعنـي ان $p = q = T \quad or \quad F$) ومـن ثـم فـان $p \rightarrow q$ او $p \rightarrow q$ كلاهـما صـحيحة True وبعكسه (*conversely*) فانه اذا $p \rightarrow q$ او $q \rightarrow p$ كلاهما صحيحة فان $q \Leftrightarrow p$ هي صحيحة ايضاً.(لاحظ المثال 1-18) . اننا مـن المحتمـل ان نلاحـظ التشـابه بـين الرمـز $\Rightarrow$ للدلالـة ضـمنا "*for implies*" والرمـز $\rightarrow$ للجمـل الشرطية (فقط اذا , و اذا ثم) "*if only , if then*" . فانه من الطبيعي ان نقرأ $p \rightarrow q$ كـ "*p implies q*" بالاضـافة الى قرائتها "*p only if q*".

نحن نختم هذا الجزء بتعريف معكـوس النفـي (*Contra positive*) للعبـارات الشرطية حيـث سـنرى لاحقـا انـه صـيغة مكافئة منطقيا وصيغة بديلة للعبارة الشرطية كما سنرى في التعريف الاتي:.

تعريف 1-7

ان معكوس النفي (*Contra positive*) للعبارة الشرطية $p \rightarrow q$ هو العبارة $\bar{p} \rightarrow \bar{q}$ ويرمز لها بشكل اخـر هـو ($\neg p \rightarrow \neg q$) . لاحظ الفرق بين معكوس النفي والمعكوس لعبارة ما . ان معكوس الجملة الشرطية هـو عكـس الادوار للعبارة p و العبارة q. اي اصبحت $p \rightarrow q$ بالصيغة المعكوسة بـ $q \rightarrow p$ لكن معكوس النفـي هـو عكـس ادوار العبارة p و العبارة q اولاً ثم نفي كل منهما واصبحت بالصيغة ($\neg p \rightarrow \neg q$) او بالصيغة $\bar{p} \rightarrow \bar{q}$.

مثال 1-19:

اكتـب العبـارة التاليـة : $if \quad 1<4, \quad then \quad 5>8$ رمزيـاً (*Symbolically*) . اكتـب المعكـوس (*the converse*) ومعكوس النفي (*The contra positive*) رمزياًً وبالكلمات ثم اوجد القيمة الحقيقية *Truth value* لكل عبارة.

الحل:

اذا عرفنا : $p \quad :1<4, \quad q :5>8$ فان العبارة المعطاة " $if \quad 1<4, \quad then \quad 5>8$ " تكتب رمزياً بـ:

$$p \rightarrow q$$

وتكتب بشكل معكوس (*converse*) بـ

$$q \rightarrow p$$

وتكتب بشكل كلمات بـ

$$if \quad 5>8, \quad then \quad 1<4$$

وتكتب بشكل نفي المعكوس بـ

$$\bar{q} \rightarrow \bar{p}$$

وتكتب بشكل كلمات بـ

If 5 is not greater than 8, then 1 is not less than 4.

ونرى مما ذكر اعلاه ان العبارة $p \rightarrow q$ هي عبارة خاطئة , وان العبارة $q \rightarrow p$ هـي عبـارة صـحيحة , واخـيرا فـان $\bar{q} \rightarrow \bar{p}$ هي عبارة خاطئة . ▲

نظرية 1:

ان العبارة الشرطية $p \rightarrow q$ ونفي معكوسها (*its contra positive*) $\bar{q} \rightarrow \bar{p}$ هما متكافئتان منطقياً. كما هو واضح في الجدول(1-17) التالي:

p	q	$p \rightarrow q$		$\bar{q} \rightarrow \bar{p}$
T	T	T		T
T	F	F		F
F	T	T		T
F	F	T		T

جدول (1-17)

ملاحظة: برهن الجدول اعلاه اي اثبـت ان $p \rightarrow q$ و $\bar{q} \rightarrow \bar{p}$ هـما متكافئـان منطقيـاً مسـتخدماً جـدول القـيم (*Truth value*) .

مـن جانـب اخـر ان كتابـة جـداول القـيم للعبـارات الاربعـة $p \rightarrow q$, $\bar{q} \rightarrow \bar{p}$, $q \rightarrow p$ و العبـارة $\neg p \rightarrow \neg q$ (*contraconverse*) نجــد ان العبــارة الشرــطية $p \rightarrow q$ ونفــي معكوســها (*contrapositive*) $\bar{q} \rightarrow \bar{p}$ هما عبارتان متكافئتان ونجـد ان المعكـوس (*converse*) $q \rightarrow p$ ومنفـي معكوسـه (*contraconverse*) للعبـارة الشرـطية هـما مكافئـان لبعضـهما الاخـر لكـنهما لـيس مكافئـان للعبـارة الشرـطية الاصـلية $p \rightarrow q$ لاحظ الجدول (1-18) التالي:

Conditional contrapositive converse contraconverse

p	q	$p \to q$	$\neg q \to \neg p$	$q \to p$	$\neg p \to \neg q$
T	T	T	T	T	T
T	F	F	F	T	T
F	T	T	T	F	F
F	F	T	T	T	T

جدول (1-18)

1-11 قوانين التوزيع Distributed Laws

تشتمل المتكافئات على مايلي:

1- $p \wedge (q \vee r) \Leftrightarrow (p \wedge q) \vee (p \wedge r)$

وهذا قانون التوزيع للاداة المنطقية AND (∧) على الاداة المنطقية OR (∨). ان هذا شبيه بما نستخدمه بالرياضيات عندما نكتب:

$$p.(q + r) = p.q + p.r$$

2- $p \vee (q \wedge r) \Leftrightarrow (p \vee q) \wedge (p \vee r)$

وهذا قانون التوزيع للاداة المنطقية OR (∨).على الاداة المنطقية AND (∧).

1-12 قوانين الترابط Associative Laws

1- $p \vee (q \vee r) \Leftrightarrow (p \vee q) \vee r$

وهذا قانون الربط للاداة المنطقية OR(∨)

2- $p \wedge (q \wedge r) \Leftrightarrow (p \wedge q) \wedge r$

وهذا قانون الربط للاداة المنطقية AND (∧).

13-1 عمليات التبادل Commutative Operations

1- ان العملية الرياضية التالية صحيحية لكل قيم x و y :

$$x + y = y + x \text{ for all } x \text{ and } y \text{ values}$$

اي تبديل موقع x بـ y او العكس لايؤثر على النتيجة .اي ان عملية الجمع هي عملية تبادل " *commutative operation* " لكن عملية الطرح ليست كذلك لأن :

$$because\ 4 - 2 \neq 2 - 4.$$

وعلى اساس ذلك فان الادوات المنطقية $\oplus$, $\vee$, $\wedge$ **هي** عملية تبادل . على سبيل المثال :

$$p \wedge q = q \wedge p$$

2- اذا كانت Δ تمثل اداة منطقية ما, فان الاداة هذه تكون ترابطية "*associative*" اذا كانت الجملتين التاليتين :

$$1 \text{-}\ x\Delta(y\Delta z) \qquad 2 \text{-}\ (x\Delta y)\Delta z$$

متكافئين منطقياً لكل قيم *z, y, x* . وعلى هذا الاساس تكون الادوات المنطقية ($\wedge, \ \vee, \ \Leftrightarrow, \ and \oplus$) هي ترابطية . على سبيل المثال :

$$x \wedge (y \wedge z) \Leftrightarrow (x \wedge y) \wedge z$$

لاحظ الاداة المنطقية AND ($\wedge$) وقارنها بالرمز Δ اعلاه.

3-عملية الضرب هي عملية توزيع لعملية الجمع كما في المثال التالي:

$$x.(y + z) = x.y + y.z$$

لكن عملية الجمع ليست عملية توزيع على عملية الضرب لأن :

$$x + (y.z) \neq (x + y).(x + z)$$

لكل قيم *z, y, x* .

لذا اذا كانت Δ تمثل اداة منطقية ما بين معاملين "*two operands*" , فانه يقال عنها انها اداة توزيع اذا تحقق مايلي:

$$p\Delta(q \circ r) \Leftrightarrow (p\Delta q) \circ (p\Delta r)$$

وباستخدام هذا الاثبات وجدول القيم الحقيقية "*truth table*"نستطيع ان نبين ان الادوات $\vee$, $\wedge$, تتوزع على بعضها الاخر . بينما الادوات $\Rightarrow$ and , $\vee$, $\wedge$ تتوزع على نفسها.

14-1 العبارات المُقدَّرة QUANTIFIERS:

ان المنطق الذي استخدم مع العبارات فيما سبق غير قادر على وصف اغلب الجمل (*Statements*) في الرياضيات وعلم الحاسوب. اعتبر على سبيل المثال الجملة التالية :

p: *n is an odd integer*

وكما قلنا ان العبارة (*proposition*) هي جملة (*statement*) و تكون اما صحيحة او خاطئة. فان الجملة المذكورة هي ليست عبارة بسبب ان قيمة p غير محددة بانها صحيحة او خاطئة لأن ذلك يعتمد على قيمة n . على سبيل المثال ان p هي *True* عندما قيمة n هي 103 (فردي) وتكون جملة خاطئة اذا كانت n هي 8 (زوجي). وبما ان معظم الجمل في الرياضيات وعلم الحاسوب يستخدمون المتغيرات (*variables*) , اذن يجب علينا ان نوسع نظام المنطق ليشمل تلك الانواع من الجمل.

تعريف 1-8

افرض ان $P(x)$ هي جملة تشتمل على المتغير x وان D هي مجموعة Set . فاننا نسمي P دالة ضمنية " *Propositional function* " فيما يتعلق بـ D اذا كانت لكل قيمة لـ x في D , انَ $P(x)$ تصبح عبارة (*proposition*) . ان D تدعى هنا بالمجال (Domain) لـ P. ولتوضيح ذلك نأخذ المثال التالي:

مثال 1-20

افرض ان $P(n)$ الجملة التالية :

n is an odd integer.

وافرض ان D هي مجموعة الاعداد الصحيحة الموجبة (*positive integers*) . ثم P هي دالة ضمنية بمجال D حيث لكل قيمة لـ n في D ان $P(n)$ هي عبارة (علي سبيل المثال لكل قيمة لـ n في D , ان $P(n)$ هي اما True او False).

فمثلا اذا كانت $n = 1$ نحصل على العبارة :

1 is an odd integer.

التي هي صحيحة . لكن اذا كانت $n = 2$ نحصل على العبارة :

2 is an odd integer.

وهي خاطئة .

ان الدالة P بنفسها لا هي صحيحة ولا خاطئة . ومع ذلك فان لكل قيمة لـ x ضمن مجالها ان $P(x)$ هي عبارة اي اما صحيحة او خاطئة .

ونستنتج من ذلك ان الدالة الضمنية " *Propositional function*" هي كتعريف لعدة عبارات (*class of propositions*) حيث تكون عبارة واحدة لكل عنصر من عناصر مجالها .على سبيل المثال لو كانت P هي دالة ضمنية بمجال مساوي الى مجموعة الاعداد الصحيحة الموجبة فاننا نحصل على العبارات :

$$P(1), P(2).........$$

وكل من $P(1), P(2)$ هو اما True او False . ▲

مثال 1-21

بيّن ان مايأتي يمثل دوال ضمنية " *Propositional function*":

-1 " n^2 *+2n is an odd integers*" ومجال الدالة هو مجموعة الاعداد الصحيحة الموجبة.

وهنا لكل عدد صحيح موجب n نحصل على عبارة (اي جملة لها قيمة صحيحة او خاطئة) , فمثلا لو $n = 1$ نحصل على *3 is an odd integer.* وهي صحيحة ولو كانت $n = 2$ نحصل على :

8 is an odd integer وهي خاطئة وهكذا.اذن " n^2 *+2n is an odd integers*" هي دالة ضمنية **حسب التعريف 1-8** الانف الذكر .

-2 " x^2*-x-6=0* " ومجال الدالة هو مجموعة الاعداد الحقيقية *real numbers* .

وبشكل مشابه للفقرة 1 , فانه لكل عدد حقيقي x نحصل على عبارة (اي جملة لها قيمة صحيحة او خاطئة) , فمثلا لو كانت $x = -2$ فان الجملة " x^2*-x-6=0* " هي صحيحة ولو كانت $x = 1$ فانها خاطئة . اذن " x^2*-x-6=0* " هي دالة ضمنية حسب **التعريف1-** 8.

تعريف 1-9

افرض ان P هي دالة ضمنية " *Propositional function*" بمجال يتحدد بـ D . فان الجملة:

for every x, P(*x*)

نسميها جملة معدودة او مُقدَّرة شاملة (عمومية) (*Universally Quantified Statement*) . الرمز $\forall$ يعني "*for every*". لذا ستكون الجملة " *for every x, P*(*x*) " تكتب بالشكل التالي:

$$\forall x P(x)$$

ان الرمز $\forall$ يدعى بـ المحدد العام *Universal Quantifier* . وستكون الجملة

for every x, $P(x)$

صحيحة اذا $P(x)$ هو صحيح لكل قيم x في D . وستكون الجملة

for every x, $P(x)$

خاطئة اذا $P(x)$ خاطئة لقيمة واحدة لـ x على الاقل في D .

ان الجملة :

for some x, $P(x)$

يقال انها جملة محددة موجودة (*Existentially Quantified Statement*). الرمز $\exists$ يعني *"for some"* " .لذا ستكون الجملة *" for every x*, $P(x)$ " تكتب بالشكل التالي:

$$\exists x P(x)$$

ان الرمز $\exists$ يدعى بـ غير المحدد الموجود *Existential Quantifier* . وستكون الجملة

for some x, $P(x)$

صحيحة اذا $P(x)$ هو صحيح لقيمة واحدة لـ x على الاقل في D . وستكون الجملة

for some x, $P(x)$

خاطئة اذا $P(x)$ خاطئة لكل قيمة لـ x في D .

اننا نسمي المتغير x في الدالة الضمنية (*Propositional Function*) " $P(x)$ " بمتغير حر (*free variable*) والفكرة ان x هي حرة ضمن المجال المحدد للدالة . اما المتغير x في الجملة التخمينية العمومية $\forall x P(x)$ والجملة التخمينية الموجودة $\exists x P(x)$ بالمتغير المحدد المرتبط بهما (*Bound Variable*) اي ان x محددة بواسطة $\forall$ و $\exists$.

اذن فيما تقدم اشرنا بان الدالة الضمنية ليس لها قيمة حقيقية *Truth value* بينما يكون ذلك للجمل المحددة (*Quantified Statements*) . وملخص ذلك ان الجملة ذات المتغير الحر (غير المحدد *Unquantified*) هي ليست عبارة وان الجملة بدون المتغيرات الحرة ((*no unquantified*) هي عبارة .

وهناك طرق مختلفة لكتابة الجمل المحددة وغير المحددة السابقة وكمايلي:

الجملة :

for every x, $P(x)$

يمكن كتابتها :

for any x, $P(x)$ و *for all x*, $P(x)$

وان الرمز $\forall$ يقرأ *"for every "* *" for all "* او *"for any "* .

وهي تعني لكل وللكل ولأي قيمة .

من جانب اخر فان الجملة: *for some x, P(x)* يمكن كتابتها

for at least one x , P(x)

و *there exist x such that ,P(x)*

وان الرمز $\exists$ يقرأ *"for some "* *" for at least one "* او *"there exists "*

وهي تعني لبعض و على الاقل قيمة واحدة او توجد هناك قيمة واحدة.

بعض الاحيان لوصف مجال الدالة D يمكننا كتابة الجملة التخمينية العمومية (*universally quantified statement*) كمايلي :

for every x in D, P(x)

ويمكننا كتابة الجملة المحددة (*existentially quantified statement*) كمايلي:

for some x in D, P(x)

مثال 1-22

الجملة التالية:

$$\textit{for every real number } x, x^2 \geq 0$$

هي جملة معدودة شاملة (عمومية) (*universally quantified statement*) . ان مجال الجملة الذي نتحدث عنه هو مجموعة الاعداد الحقيقية . ان الجملة اعلاه ستكون صحيحة *True* بسبب ان لكل عدد حقيقي ***x*** , ان مربع ***x*** هو عدد موجب او صفر وهذا *True* . ▲

مثال 1-23

الجملة المعدودة الشاملة (العمومية) (*universally quantified statement*) التالية :

$$\textit{for every real number } x, \textit{if} \quad x > 1, \quad \textit{then} \quad x + 1 > 1$$

هي *True* . في هذا الوقت يجب علينا ان نثبت ان الجملة :

$$\textit{if} \quad x > 1, \quad \textit{then} \quad x + 1 > 1$$

هي صحيحة لكل عدد حقيقي ***x*** . ▲

الاثبات:

افرض ان x هي اي عدد حقيقي كيفما يكون , فسيكون صحيحا ان لكل عدد حقيقي x , اما $x \le 1$ او $x > 1$ تكون صحيحة . ففي حالة ان x اصغر او تساوي 1 ($x \le 1$) ان العبارة الشرطية "*the conditional proposition*" التالية :

$$if \quad x > 1, \quad then \quad x + 1 > 1$$

ستكون صحيحة بسبب ان الفرضية hypothesis ($x > 1$) هي خاطئة *False* لأننا اخذنا قيم x , ($x \le 1$) وليس $x > 1$. تذكر انه عندما الفرضية خاطئة ان الجملة الشرطية صحيحة بغض النظر عما لو كانت النتيجة صحيحة ام خاطئة . انظر الجدول (1-19) ادناه:

p	*q*	*p*	→	*q*
T	T		T	
T	F		F	
F	T		T	
F	F		T	

جدول (1-19)

الان افرض ان x اكبر من واحد ($x > 1$) , فبغض النظر عن القيمة المحددة لـ x فان قيمة x مضافاً لها واحد هي اكبر من قيمة x نفسها اي ($x + 1 > x$). وبما ان :

$$x + 1 > x \text{ and } x > 1$$

نستنتج من ذلك ان $x + 1 > 1$ وبذلك ستكون النتيجة (*conclusion*) صحيحة . واذا كانت $x > 1$ فان الفرضية والنتيجة كلاهما صحيح . اذن العبارة الشرطية :

$$if \quad x > 1, \quad then \quad x + 1 > 1$$

هي عبارة صحيحة .

من جانب اخر , لقد بينا ان لكل عدد حقيقي x , ان العبارة :

$$if \quad x > 1, \quad then \quad x + 1 > 1$$

صحيحة لذلك فان الحملة التخمينية العمومية :

$$\textit{for every real number } x, if \quad x > 1, \quad then \quad x + 1 > 1$$

ستكون صحيحة ايضا.

ان المثال – 1-23 اعلاه يزودنا بمزيد من الحافز لتعريف العبارة الشرطية $p \to q$ في ان تكون صحيحة عندما p خاطئة . فلاجل ان تكون الجملة التخمينية العمومية :

$$\text{for every real number } x, if \quad x>1, \quad then \quad x+1>1$$

صحيحة فانه يجب ان تكون العبارة الشرطية :

$$if \quad x>1, \quad then \quad x+1>1$$

بغض النظر عما تكون عليه قيمة x . وبشكل خاص فان العبارة :

$$if \quad x>1, \quad then \quad x+1>1$$

يجب ان تكون صحيحة اذا كانت ($x>1$) خاطئة .

وبناءاً على التعريف 1- 9 الانف الذكر

فان الجملة المعدودة الشاملة (العمومية) (*universally quantified statement*) التالية :

$$\text{for every } x, P(x)$$

هي خاطئة اذا على الاقل قيمة واحدة لـ x في المجال *Domain* , ان العبارة $P(x)$ تصبح عبارة خاطئة وان قيمة x في المجال التي تجعل العبارة $P(x)$ خاطئة تدعى بـ *counterexample* الى الجملة :

$$\text{for every } x, P(x)$$

1-15 العناصر التي تجعل الجمل خاطئة : *Counter Examples*

ان عنصر المجموعة الكلية لجملة حول تلك المجموعة والذي يجعل تلك الجملة خاطئة نسميه بـ ***counter example* تلك الجملة .**

مثال 1-24

اوجد العنصر "*counter example*" للجمل التالية حول الاعداد الصحيحة الموجبة n :

1- ($2n^2 + n = n^3 - n^2 + 3n$)

2- اذا كانت n عدد اولي , فان 12 و $n^3 - n^2 + n$ () يكون لهما عامل مشترك "*common factor*" .

الحل:

ان العناصر التي لاتحقق شروطنا في الجمل اعلاه هي عبارة عن القيم لـ n وهي :

$$1- \quad n=3$$

$$2- \quad n=7$$

اي انه عندما نعوض قيم (*n=1 and 2*) في الجملة الاولى فانها تتحقق . لكن عندما نعوض *n=3* فان الجملة لن تتحقق.اذن ***n=3*** هي الـ "*counter example* "

اما بالنسبة للجملة الثانية : لان n عدد اولي كشرط , فان الاعداد الاولية 3, 5, 7, 11, ... ليست كلها تحقق الجملة (العامل المشترك) . لانه فقط الاعداد 3 و 5 تحقق اما عندما نعوض *n=7* فتصبح الجملة خاطئة.

.اذن ***n***=7 هي الـ "*Counter Example* ". ▲

مثال 1-25

الجملة المحددة العمومية (الشاملة) (*the universally quantified statement*) التالية :

لاي عدد حقيقي x ان (x^2-1) اكبر من 1 .

for every real number x , $x^2-1>0$

جملة خاطئة بسبب انه اذا x=1 , فان العبارة :

$$1^2-1>0$$

هي عبارة خاطئة . وتكون القيمة 1 هي الـ counterexample الى الجملة :

for every real number x , $x^2-1>0$

ولبيان ان الجملة المعدودة الشاملة (*universally quantified statement*) التالية :

for every x, P(x)

هي خاطئة, فانه يكفي ان نجد قيمة واحدة لـ *x* في المجال *Domain* , ان تكون فيه العبارة $P(x)$ خاطئة . ان الطريقة لأثبات بطلان الجملة:

for every x, P(x)

تختلف تماما من الطريقة المستخدمة لبرهنة ان الجملة صحيحة .ولأثبات ان الجملة:

for every x, P(x)

صحيحة , فانه يجب في الواقع ان نختبر كل قيمة لـ x في المجال الذي نتحدث عنه ونبين انه لكل x , ان العبارة $P(x)$ صحيحة . ▲

مثال 1-26

الجملة المحددة او المعدودة الشاملة " *universally quantified statement* " التالية :

"لكل عدد صحيح موجب n , اذا كانت n عدد زوجي , فان (n^2+n+19) تصبح عدد اولي "*prime*"

$$\text{for every positive integer n, if n is even , then } n^2+n+19 \text{ is prime}$$

هي جملة خاطئة False . وقيمة n التي لاتحقق الشرط "*counterexample*" تحصل عندما ناخذ n=38 . اذن العبارة الشرطية "*conditional proposition*" التالية:

$$\text{if 38 is even , then } 382+38+19 \text{ is prime}$$

هي جملة خاطئة لأن الفرضية "*hypothesis*" التالية:

$$\text{38 is even}$$

هي صحيحة . لكن النتيجة "*conclusion*" التالية :

$$38^2+38+19 \quad \text{is prime}$$

هي خاطئة لأن حاصل الجمع هو ليس عددا اولياً: $38^2+38+19 \quad \text{is} \quad \text{not} \quad \text{prime}$

وكما يلي:

$$▲\ 38^2+38+19 = 38.38+38+19 = 19(2.38+2+1) = 19.79$$

نعود مرة ثانية الى الجمل المعدودة الموجودة " *existentially quantified statements*". طبقا الى التعريف **1-9** , فان الجملة المحددة الموجودة التالية:

$$\textit{for some x in D,P(x)}$$

هي صحيحة اذا $P(x)$ هي صحيحة لقيمة واحدة على الاقل لـ x في المجال D . اذا $P(x)$ هي صحيحة لبعض قيم x فانه ممكن وبشكل مؤكد ان $P(x)$ هي خاطئة لقيم x الاخرى.

مثال 1-27

الجملة المحددة (المعدودة) الشاملة " *universally quantified statement* " التالية :

$$\text{for some real number } x, \frac{x}{x^2+1}=\frac{2}{5}$$

هي صحيحة بسبب انه من الممكن ايجاد على الاقل قيمة حقيقية واحدة لـ x تجعل العبارة $\frac{x}{x^2+1}=\frac{2}{5}$ صحيحة.

فعلى سبيل المثال اذا كانت قيمة $x=2$ فاننا نحصل على العبارة الصحيحة :

$$\frac{2}{2^2+1}=\frac{2}{5}$$

لكن ليست كل قيم x تنتج عبارة صحيحة . مثلا العبارة :

$$\frac{1}{1^2+1}=\frac{2}{5}$$

هي خاطئة حيث قيمة $x=1$. ▲

مثال 1-28

الجملة المعدودة (المُقدّرة) الشاملة " *universally quantified statement* " التالية :

"لبعض قيم عدد صحيح موجب n , اذا n هو عدد اولي , فان ($n+1$ و $n+2, n+3, n+4$) ليست اعداد اولية "

" for some positive integer n, if n is prime ,then n+1,n+2,n+3 and n+4 are not prime"

ان الجملة اعلاه صحيحة بسبب انه من الممكن ايجاد على الاقل قيمة صحيحة واحدة لـ n تجعل العبارة :

" if n is prime ,then n+1,n+2,n+3 and n+4 are not prime"

صحيحة . على سبيل المثال , اذا كانت $n=23$, فنحصل على العبارة الصحيحة التالية:

if 23 is prime, then 24, 25, 26, and 27 are not prime

تلك العبارة الشرطية هي صحيحة بسبب ان كل مـن الفرضـية *hypothesis* " 23 is prime " والنتيجـة *conclusion* "24, 25, 26, و 27 " صحيحاً. ان بعض القيم لـ n تجعل العبارة الشرطية صحيحة (مـثلاً :n=4, n=23, n=47) بيـنما الـبعض الاخر يجعلها خاطئة : (مثلاً: (n=2, n=101) .

النقطة التي نبينها هنا اننا قد وجدنا قيمة واحدة جعلت العبارة الشرطية :

" *if n is prime ,then n+1,n+2,n+3 and n+4 are not prime*"

عبارة صحيحة ولهذا السبب فان الجملة التخمينية المحددة :

" *for some positive integer n, if n is prime ,then n+1,n+2,n+3 and n+4 are not prime*"

هي صحيحة. ▲

وبناءاً على التعريف 1- 9 الانف الذكر

فان الجملة المعدودة الشاملة العامة (*universally quantified statement*) التالية :

for some x, $P(x)$

هي خاطئة اذا لكل قيمة لـ x في المجال "*Domain*" الذي نتحدث عنه , ان العبارة $P(x)$ خاطئة.

مثال 1-29

اثبت ان الجملة المعدودة الشاملة العامة (*universally quantified statement*) التالية :

" لبعض القيم الحقيقية لـ x , ان $\frac{1}{x^2+1} > 1$."

$$\text{for some real number } x,\ \frac{1}{x^2+1} > 1$$

هي خاطئة:

لأثبات ذلك يجب علينا ان نبين ان " $\frac{1}{x^2+1} > 1$ " خاطئة لكل عدد حقيقي x . لذا ستكون " $\frac{1}{x^2+1} > 1$ "

خاطئة بدقة, اذا كانت " $\frac{1}{x^2+1} \leq 1$ " صحيحة. لذا يجب علينا ايضا ان نبين ان " $\frac{1}{x^2+1} \leq 1$ " صحيحة لكل عدد حقيقي x . لغاية هنا افرض x ان تكون اي عدد حقيقي , بما ان الصفر اصغر او يساوي مربع x ($0 \leq x^2$) , وباضافة (1) الى طرفي هذه المتباينة "*inequality*" " $0 \leq x^2$ " نحصل على:

$$1 \leq x^2 + 1$$

واذا قسمنا طرفي تلك المتباينة بـ " x^2+1 " نحصل على :

$$\frac{1}{x^2+1} \leq 1$$

لذلك ستكون الجملة " $\frac{1}{x^2+1} \leq 1$ " صحيحة True لأي عدد حقيقي x . وبالتالي فان الجملة : $\frac{1}{x^2+1} > 1$ خاطئة لكل عدد حقيقي x .

▲

واخيرا فنحن بينا ان الجملة المعدودة المحددة "*existentially quantified statement* " التالية

$$\text{for some } x, \frac{1}{x^2+1} > 1$$

هي خاطئة.

في هذا المثال وضحنا ان الجملة المعدودة المحددة " *existentially quantified statement* " خاطئة وذلك بواسطة برهاننا ان الجملة المعدودة الشاملة المتعلقة بها " *related universally quantified statement* " صحيحة . وفي النظرية القادمة ستكون العلاقة واضحة ودقيقة .وهذه النظرية تضع قوانين موركان للمنطق " "*De Morgan's laws for logic*" كما اشرنا اليها سابقا في المثال (1-16).

16-1 نظرية 2 : تعميم قوانين دي موركان للمنطق *Generalized De Morgan Laws For Logic*

اذا كانت ***p*** دالة افتراضية(ضمنية) "*propositional function* " فان اي زوج من العبارات في (a) و (b) ادناه يكون لـه نفس القيم الحقيقية *"the same truth values"* (على سبيل المثال اما كلاهما صحيح True او كلاهما خاطئ False).

$(a) \quad \overline{\forall x P(x)}; \exists x\, \overline{p(x)}$

$(b) \quad \overline{\exists x P(x)}; \forall x\, \overline{p(x)}$

ولبرهنة ذلك :فاننا سنبرهن الجزء (a) ونترك الجزء (b) للقارئ.

البرهان:

افرض ان العبارة $\overline{\forall x P(x)}$ هي صحيحة . اذن العبارة $\forall x P(x)$ هي خاطئة لانها النفي للعبارة الاولى . وبواسطة التعريف1- 9 فان العبارة $\forall x P(x)$ خاطئة وبشكل دقيق عندما $P(x)$ هي خاطئة لقيمة واحدة لـ x على الاقل ضمن المجال الذي تأخذه x . لكن اذا $P(x)$ هي خاطئة لقيمة واحدة لـ x على الاقل ضمن المجال الذي نتحث عنه فان نفيها $\overline{P(x)}$ يكون صحيحا على الاقل لقيمة واحدة لـ x ضمن المجال المشار اليه . وبالرجوع مرة اخرى الى التعريف 1- 9 فانه عندما $\overline{P(x)}$ هي صحيحة لقيمة واحدة لـ x على الاقل ضمن المجال المذكور , فان العبارة $\exists x\, \overline{p(x)}$ هي صحيحة. وهذا يعني انه عندما العبارة $\overline{\forall x P(x)}$ صحيحة اثبتنا ان العبارة $\exists x\, \overline{p(x)}$ هي صحيحة ايضا. وبشكل مشابه يمكننا ان نثبت انه اذا كانت العبارة $\overline{\forall x P(x)}$ خاطئة فان العبارة $\exists x\, \overline{p(x)}$ هي ايضا خاطئة . لذلك فان زوج العبارات في (a) دائما له نفس القيم الحقيقية " *truth values* "

مثال 1-30

افرض ان $P(x)$ هي الجملة :

$$\frac{1}{x^2+1} > 1$$

ففي المثال (1-27) بينا ان العبارة :

$$\textit{for some real number } x,\ P(x)$$

هي خاطئة باثبات ان :

$$\textit{for every real number } x,\ \overline{P(x)}$$

هي صحيحة .

التكنيك هنا يمكن صياغته بالرجوع الى **نظرية 2: تعميم قوانين دي موركان للمنطق** السالفة الذكر :

فبعد ان برهنا ان :

$$\textit{for every real number x}, \ \overline{P(x)}$$

هي صحيحة , علينا ان ننفيها ونستنتج ان :

$$\overline{\text{for every real number x}, \overline{p(x)}}$$

هي خاطئة كمايلي:

الاثبات:

مرة ثانية بواسطة **نظرية 2: تعميم قوانين دي موركان للمنطق** الجزء (a) اعلاه :ان

$$\textit{for some real number x}, \ \overline{\overline{P(x)}}$$

او مايكافئها

$$\textit{for some real number x}, \ P(x)$$

هي خاطئة ايضاً .

ان عبارة معدودة شاملة (عامة) "*<u>a universally</u> quantified proposition*" تستقرئ العبارات المركبة "*compound proposition*" التالية :

$$P_1 \wedge P_2 \wedge ..... \wedge P_n$$

في ادراك يفهم ان العبارات الفردية "*individual propositions*" , P_1 وP_2 وP_n تبدل بواسطة $P(x)$ ، حيث x هي عنصر المجال الذي نتحدث عنه وان العبارة : " $P_1 \wedge P_2 \wedge ..... \wedge P_n$ " توضع مكان العبارة :

$$for\ every\ \ x,\ P(x)$$

ان العبـارة $P_1 \wedge P_2 \wedge ..... \wedge P_n$ هـي صـحيحة اذا وفقـط اذا (*if and only if*)، P_i (هـي صـحيحة لكـل قـيم ($i = 1,.......n.$) وبشكل مشابه يمكننا تعريف القيمة الحقيقة *Truth value* للعبارة :

$$for\ every\ \ x,\ P(x)$$

ان العبارة " for every x, P(x) "هي صحيحة اذا وفقط اذا (*if and only if*) , $P(x)$ هي صحيحة لكـل قيمـة من قيم x في المجال.
▲

وبشـكل مشـابه ان عبـارة موجـودة مقـدَرة "*<u>an existentially</u> quantified proposition*" تسـتقرئ العبـارات المركبـة "*compound proposition*" التالية :

$$P_1 \vee P_2 \vee ..... \vee P_n$$

في ادراك يفهم ان العبارات الفردية "*individual propositions*" ، P_1 و P_2 و P_n تبـدل بواسـطة $P(x)$, حيـث x هي عنصر المجال الذي نتحدث عنه وان العبارة : " $P_1 \vee P_2 \vee ..... \vee P_n$ " تبدل "*replaced*" مكان العبارة :

$$for\ some\ \ x,\ P(x)$$

ان ماتقدم يوضح كيف ان **نظرية 2 تعمم قوانين دي موركان للمنطق,**(راجع تعريف 6-1 (مثال 1-16)).

تذكر ان قانون دي موركان الاول للمنطق "*first De Morgan law for logic*" ينص على ان العبارات :

$$\overline{P_1 \vee P_2 \vee ..... \vee P_n} \qquad و \quad \overline{P}_1 \wedge \overline{P}_2 \wedge ..... \wedge \overline{P_n}$$

يكون لهما نفس القيم الحقيقية .

من جانب اخر في نظرية رقم 2 الجزء (b) ان العبارة " " $\overline{P}_1 \wedge \overline{P}_2 \wedge ..... \wedge \overline{P_n}$ تبدل بـ

$$\forall x \overline{p(x)}$$

وان العبارة " $\overline{P_1 \vee P_2 \vee ..... \vee P_n}$ " تبدل بـ

$$\overline{\exists x p(x)}$$

مثال 1-31

ان الجمل بدلالة الكلمات غالبا مايكون لها اكثر من تفسير.اعتبر الجملة المشهورة التالية:

" ليس كل مايلمع هو ذهب : *all that glitters is not gold* "

ان احد التفسيرات الممكنة لهذه الجملة هو :" لاشئ يلمع هو ذهب: *Nothing that glitters is gold* "

بمعنى اخر الذهب ليس لماعاً .مع ذلك هذا لم يكن المقصود من الجملة كما قصدها قائلها . لكـن الجـواب الصـحيح هـو: " ليس بعض ما يلمع هو ذهب : *something that glitters is not gold* ".

فاذا فرضنا ان $P(x)$ تكون الدالة الضمنية "*propositional function*" التالية :

"*x glitters*"

واذا فرضنا ان $Q(x)$ تكون الدالة الضمنية "*propositional function*" التالية"

" *x is gold* "

فان التفسير الاول يصبح مايلي:

$$for \quad all \quad x, p(x) \rightarrow \overline{Q(x)}$$

والتفسير الثاني يصبح :

$$for\ some\ x, P(x) \wedge \overline{Q(x)}$$

وباستخدام نتيجة المثال 1-17 فاننا نرى ان القيم الحقيقية " *truth values*" لـ " $for\ some\ x, P(x) \wedge \overline{Q(x)}$ "

و " $\overline{\text{for some x}, \text{P(x)} \rightarrow \text{Q(x)}}$ " هي نفسها.

وباستخدام النظرية (2) فان القيم الحقيقية لـ

$$for\ some\ x, \overline{P(x) \to Q(x)}$$

و

$$\overline{for \quad all \quad x, P(x) \to Q(x)}$$

هي نفسها ايضاً. ولهذا فان الاسلوب المكافئ لتمثيل التفسير الثاني هو :

$$\overline{for \quad all \quad x, P(x) \to Q(x)}$$

وبمقارنة هذا التمثيل مع " $for \quad all \quad x, p(x) \to \overline{Q(x)}$ " نرى ان الغموض ينتج من فيما لو استخدم النفي لـ $Q(x)$ (التفسير الاول) اواستخدم النفي لكامل الجملة : $\overline{for \quad all \quad x, p(x) \to Q(x)}$ وهو التفسير الثاني.

ان التفسير الصحيح للجملة : " ليس كل مايلمع هو ذهب : *all that glitters is not gold* "

ينتج من نفي هذه الجملة بكاملها. ▲

من جانب اخر,في جمل موجبة *"positive statements"* ان (*every* و , *each* , *all* , *any*) لها نفس المعنى . لكن في الجمل المنفية *"negative statements"* الحالة تتغير. فلو اخذنا الجمل التالية:

" ليس كل قيم x تحقق $P(x)$:. " *Not all x satisfy P(x)*

" ليست كل قيمة لـ x تحقق $P(x)$:. " *Not each x satisfies P(x)*

" ليس لكل قيمة لـ x تتحقق $P(x)$:. " *Not every x satisfy P(x)*

فان هذه الجمل تعتبر ان لها نفس المعنى وهو:

$$for \quad some \quad x, \overline{P(x)}$$

في حين ان الجملتين التاليتين:

Not any x satisfies P(x)

Not x satisfies P(x)

تعنيان : $for \quad all \quad x, \overline{P(x)}$

اخيرا في الامثلة القادمة سـنمزج فيهـا بـين العبـارات غـير المحـددة - الشـاملة والموجـودة *mix universal and existence quantifiers*" " في جملة مفردة وكذلك للقياس على اكثر من متغير " *variables* "

مثال 1-32

1- افرض ان المجال الذي نتحدث عنه هو مجموعة الاعداد الحقيقية .اعتبر الجملة :

$$for \ every \ x, \ for \ some \ y, x+y=0$$

ان معنى تلك الجملة هو "لأية قيمة لـ x مهما كانت , فانه يوجد على الاقل قيمة واحدة لـ y التي ربما تعتمد عـلى الخيـار لـ x , ان $x+y=0$."

وبالامكان ان نبين ان الجملة : $for \ every \ x, \ for \ some \ y, x+y=0$ "" هي صحيحة كمايلي:

لأية قيمة لـ x , يمكننا ان نجد على الاقل قيمة واحدة لـ y , نعني " $y=-x$ " . مثل ذلك " $x+y=0$ "

2- افرض اننا عكسنا الجملة السابقة لتقرأ :

$$for \ some \ y, \ for \ every \ x, x+y=0$$

فاذا كانت تلك الجملة صحيحة , فانه من الممكن عندما نختار بعض القيم لـ y , ان الجملة التالية تكون صحيحة :

$$for \ every \ x, x+y=0$$

ومع ذلك يمكننا ان نثبت بطلان هذه الجملة بقيمة ما "*counterexample*" . على سبيل المثال ,ربما ناخذ " $x=y-1$ " ومن ثم نحصل الجملة الخاطئة التالية:

$$1-y+y=0$$

وعلى اساس ذلك , فان الجملة :

$$for \ some \ y, \ for \ every \ x, x+y=0$$

تكون جملة خاطئة. ▲

مثال 1-33

افرض ان $P(x,y)$ تمثل الجملة:

$$if \quad x^2 < y^2, then \quad x < y.$$

وان المجال الذي نتحدث عنه هنا هو مجموعة الاعداد الحقيقية "*real numbers*" .

فالجملة التالية:

$$for\ every\ x,\ for\ every\ y, P(x,y)$$

تكون خاطئة , حيث القيم "*counterexample*" التي تسبب ذلك هي x=1, y=-2 . وفي هـذه الحالـة نحصـل عـلى العبـارة التالية:

$$\text{if } 1^2 < (-2)^2, \text{ then } 1 < -2$$

لكن من جانب اخر ان الجملة التالية:

$$for\ every\ x,\ for\ some\ y, P(x,y)$$

هي صحيحة .و يمكننا ايضا ان نبين ان لكل قيمة لـ x , ان العبارة التالية :

$$for\ some\ y, if\ x^2 < y^2,\ then\ x < y$$

تكون صحيحة, وذلك بواسطة اخذ قيمة لـ y تكون فيها العبارة :

$$"\ "if \quad x^2 < y^2, then \quad x < y$$

صحيحة. ▲

وفي الحقيقة لو وضعنا "y = 0" , نحصل على العبارة الصحيحة التالية:

$$if \quad x^2 < 0,\ then \quad x < 0$$

"ان العبارة الشرطية هذه صحيحة بسبب ان الفرضية $x^2 < 0$ "" خاطئة.

وبنفس الطريقة فان الجملة التالية :

$$for\ every\ y,\ for\ some\ x, P(x,y)$$

صحيحة . كما يمكننا ان نبين ان لكل قيمة لـ y , ان العبارة :

$$for\ some\ x, if\ x^2 < y^2,\ then\ x < y$$

صحيحة وذلك بواسطة اخذ قيمة لـ x تكون فيها العبارة :

$$if \quad x^2 < y^2 ,then \quad x < y$$

صحيحة. وفعلا لو وضعنا قيمة لـ " $x = |y| + 1$ " فاننا نحصل على العبارة الصحيحة التالية :

$$\text{if} \quad (|y|+1)^2 < y^2, \quad \text{then} \quad |y|+1 < y.$$

"ان العبارة الشرطية هذه صحيحة بسبب ان الفرضية خاطئة.

واخيرا يمكننا ان نلخص القواعد التي تحقق البرهان او تدحضه للجمل المحددة اوالشاملة الانفة الذكر كمايلي:

1- لبرهان ان الجملة المحددة الشاملة *"universally quantifier statement"* التالية :

$$for\ every\ x, P(x)$$

هي صحيحة فاننا نبين ان لكل x (*every x*) في المجال , ان العبارة P(x) هي صحيحة.حيث ان بيان P(x) هي صحيحة لـبعض القيم الخاصة بـ *x* لايبرهن ان العبارة

$$for\ every\ x, P(x)$$

هي صحيحة.

2- لبرهان ان الجملة المحددة الموجودة" *existential quantifier statement* " التالية :

$$for\ some\ x, P(x)$$

هي صحيحة فاننا نجد قيمة واحدة لـ x في المجال ,تكون فيها العبارة P(x) هي صحيحةحيث قيمة واحدة كافية .

3- لبرهان ان الجملة المحددة الشاملة التالية :

$$for\ every\ x, P(x)$$

هي خاطئة فاننا نجد قيمة واحدة لـ x (*counterexample*) في المجال , تكون فيها العبارة P(x) خاطئة.

4- لبرهان ان الجملة المحددة الموجودة" *existential quantifier statement* " التالية :

$$for\ some\ x, P(x)$$

هي خاطئة فاننا نبين ان لكل x (*every x*) في المجال , ان العبارة P(x) هي صحيحة , حيث ان بيان P(x) هي خاطئة لـبعض القيم الخاصة بـ *x* لايبرهن ان العبارة :

$$for\ some\ x, P(x)$$

هي خاطئة.   ▲

17-1 الجمل المقدرة والجمل المفتوحة (الغير مُقدِّرة) Quantifiers And Predicates

ان لغة الجملة " *statement language* " غير كافية بقوة لعمل كل التعابير الرياضية *"assertions"* التي نحتاجها في الرياضيات" *mathematics* ". مثلا اننا نحتاج الى:

$$"x=3","x\geq y","x+y=z".$$

مثل تلك التعابير ليست جملاً لأنها ليست بالضرورة اما صحيحة او خاطئة .مع ذلك اذا تم تعيين قيمها فانها تصبح جملاً. مثل تلك التعابير تحدث في اللغة ايضا كما يلي :

1- هو طويل واشقر الشعر - *He is tall and blond-* ويمكننا ان نقول *x is tall and blond*

2- هي تعيش في المدينة - *She lives in the city* - ونعبر عنها ايضا بـ . " *x lives in y* ".

ان التعبيرين " *x is tall and blond*" و " *x lives in y* ". يسميان بـ بالجمل غير المقدرة "*Predicates*" وهي لوصف العلاقة بين المتغيرات ضمن التعبير. وهنا ان التعابير تعمل بالخبريات والمتغيرات تصبح صحيحة *True* او خاطئة *False* عندما تعطى قيماً "*values*" .

وللسهولة فنحن نرمز لكل "*predicates*" على انه "*predicate* " دون الاشارة الى المتغيرات الداخلة فيها . فمثلا في التعبير التالي:

" *x is tall and blond* "

ان *x* تعتبر متغير , وان " *is tall and blond* " هي " *predicate*" , وفي التعبير . " *x lives in y* " ان *x* و *y* هما متغيريان وان " *lives in* " هي " *predicate*".

مثال 1-34

ان الجمل غير المقدرة "*predicates* " في الغالب تستخدم مع جمل التحكم "*control statements*" .

ففي لغات البرمجة عالية المستوى مثل فيجوال بيسك "*visual basic*" , لغة C ولغة كوبل Copal , ولغة فورتران Fortran ....الخ ., فمثلاً لو قلنا الجملة التالية :

$$If \quad x > 3 \ \ then \ \ y \leftarrow z$$

فان "$x > 3$" هو "*predicate* " وعندما تنفذ الجملة فان القيم الحقيقية "*Truth values*" للتعبير "$x > 3$ " تحددها القيمة الحالية لـ *x* في البرنامج . فلو تم تعيين القيمة الحقيقية (1) للتعبير فانه صحيح ولو تم تعيينها (0) فانه خاطئ . فمثلا في بعض لغات البرمجة مثل "PL/1 " ان جملة التعيين التالية :

$$\mathrm{A} = \mathrm{x} < 3$$

تعني ان (A = 1) اذا كان $x < 3$ "" صحيح " *if* $x < 3$ is true "اي قيم *x* اما 0, 1, او 2 . :

وتعني ان (A = 0) اذا كان $x < 3$ "" خاطئة " *if* $x < 3$ is false "اي قيم *x* اكبر من 2 :

1- ***A*** assigned **1** *if* $x < 3$ is true .

2- ***A*** assigned **0** *if* $x < 3$ is false.

ومثال 1-35

لو قلنا مايلي:

1- ***x*** هي انثى (*x is female*) فاننا يمكن الاشارة لها بـ ***F*(*x*)**

2- ***x*** زوجت الى y (*x married to y*) , يمكن كتابتها بـ ***M*(*x*, *y*)**

3- $x + y = z$ تكتب بـ **S(*x*, *y*, *z*)** .

وتعتبر ***F*(*x*)** , ***M*(*x* ,*y*)** و **S(x, y, z)** ثوابت خبرية " *predicates constants* " تدل على حدود خبرية خاصة.

من جانب اخر لو فلنا :

$p(\ x_1\ , x_2\ , \ldots\ldots\ldots\ldots , x_n\ )$, فان *p* هي "***predicate*** " و x_i هي متغيرات او عدة حدود لـ *p*

وتسمى بـ " *n-place predicates* ". ▲

1-17-1 في منطق الجملة المفتوحة (الغير مُقدِّرة) *Predicate logic*:

من اجل ان نغير الـ "*predicate*" الى جملة فانه لابد من ان يتم حصر "*bound*" كل متغير فيه وهذا يتم بطريقتين :

1- بواسطة تعيين قيمة لكل متغير على انفراد في الـ "*predicate*" , فعلى سبيل المثال: لو اخذنا الخبر التالي :

$$x + y = 3$$

والذي يشار له بـ $p(x,y)$, فلو اعطينا $x = 1 \ , y = 2$ فان $p(x,y)$ صحيح ويتحول الى الجملة $p(1,2)$ وقيمتها الحقيقية صحيحة *True* لان : $3 = 1 + 2$, لكن لو اعطينا $x = 2 \ , y = 6$ فان $p(x,y)$ خاطئ ويتحول الى الجملة $p(2,6)$ وقيمتها الحقيقية خاطئة *false* لان : $3 \neq 2 + 6$.

2- بواسطة تخمين المتغير هل هو شامل (*Universal*) او محدد موجود(*Existential*).

2-17-1 المُقَدِر الشامل Universal Quantifier

ففي حالة الشامل (universal) , افرض ان $p(x)$ هو predicate بدلالة متغير واحد هو **x** , لذا فان التعبير :

"*for all*, $p(x)$" يكتب رمزيا كمايلي:

$$\forall x p(x)$$

وهذا يعني لكل قيم x , ان التعبير $p(x)$ هو صحيح "True". وهذا يعني ايضا : اذا كانت p(x) صحيحة فان $\forall x p(x)$ صحيحة ايضاً. وبهذا يكون x قياس شامل وان $\forall$ يمثل معنى كثير وشامل ويقرأ :

$$\text{"for all", "for any", and " for every".}$$

مثال 1-36

1- لكل قيم x , فان قيمة x اقل من $x + 1$,

اي : $\forall x[x < x + 1] : \text{for all } x, x < x + 1$. وهنا اذا كانت المجموعة الشاملة تمثل الاعداد الصحيحة فان : $x < x + 1$ يكون صحيحاً وهو مطابق لـ $\forall x$.

2- بينما لو اخذنا : لأية قيمة لـ x , ان $x = 3$ اي ($\forall x[x=3] : for\ any\ x, x=3$) ولنفس مجموعة الاعداد الصحيحة *integers* , فان $x = 3$, خاطئة لقيم x التي اقل من 3 واكبر منها وهو عكس $\forall x$. لذا تعتبر الجملة الاولى صحيحة والثانية خاطئة .

من جانب اخر لو كانت A تمثل مصفوفة اعداد صحيحة (*integer Array*) مكونة من خمسين عنصراً (A[0],A[1],A[2]...............A[50] , فانه بالمكان التعبير عنها في حالة كون العناصر لاتساوي صفراً كمايلي:

$$\forall i\{(1 \le i \wedge \le 50) \Rightarrow A[i] \ne 0\}$$

ولو خزنت هذه العناصر بطريقة ترتيب غير تنازلية "*non-decreasing order*" فاننا نعبر عن ذلك بمايلي:

$$\forall i\{(1 \le i \wedge \le 50) \Rightarrow A[i] \le A[i+1]\}$$

كذلك يمكننا ان نستخدم $\forall$ اكثر من مرة مع اكثر من متغير كما في المثال التالي:

مثال 1-37

لكل قيم x ولكل قيم y , فان مجموع x و y هو اكبر من x .

$$for\ \ all\ \ x\ \ and\ \ all\ \ y, x+y\ \ greater\ \ than\ \ x$$

$$\forall x \forall y[x+y > x]$$

وهذه الجملة صحيحة عندما تكون المجموعة الشاملة هي الاعداد الصحيحة الموجبة . بينما تكون خاطئة اذا كانت المجموعة الشاملة هي الاعداد الصحيحة فقط (اي الموجبة والسالبة).

1-17-3 المُقدَّر الموجود(المُحدِّد) Existential Quantifier

لو قلنا توجد قيمة لـ x يكون فيها التعبير $p(x)$ صحيحاً اي (*for some x*, $p(x)$) فاننا نعبر عن ذلك بمايلي:

$$\exists x p(x)$$

وتعني: *there exists a value of x for which the assertion p(x) is true*

وفي هذه الحالة لو التعبير $p(x)$ صحيحاً لقيمة واحدة لـ x على الاقل في المجموعة الشاملة التي نتحدث عنها, فان $\exists x p(x)$ صحيحة ايضاً. وخلافاً لذلك فهي خاطئة .

مثال 1-38

1- ان $\exists x[x < x+1]$ تعني انه توجد x بحيث ان x هي اصغر من x+1 :

there exists an x such that x is less that x + 1

وان $\exists x[x = 3]$ تعني انه توجد x بحيث انه x=3 :

there exists an x such that x = 3.

ان كلا الجملتين اعلاه صحيحا اذا اخذنا مجموعة الاعداد الصحيحة . لكن لو كانت الجملة الاولى:

$\exists x[x = x+1]$, فانها تصبح خاطئة لانه لا أهمية لاية قيمة تعطى لـ x هنا , حيث التعبير " *x=x+1* "يكون خاطئاً.

▲

مثال 1-39

1- استخدم $s(x)$ لتقوم مقام " $x^2 = 16$ " ,ثم اكتب الجملة التالية:

" توجد x , الى حد ان مربعها يساوي 16"

" there is a x such that " $x^2 = 16$ *"*

رمزياً.

والحل هو : $\exists x(s(x))$. ▲

مثال 1-40

مستخدما المختصرات التالية :

1-اذا كانت $s(x)$ تقوم مقام الجملة : x عدد زوجي صحيح *" x is an even integer"*

2- اذا كانت $t(x, y)$ تقوم مقام الجملة $x = 2y$.

3- اذا كانت $p(x)$ تقوم مقام الجملة : x عدد اولي صحيح *" x is prime integer"*

4- اذا كانت $r(x)$ تقوم مقام الجملة $x > 2$.

اعد كتابة الجمل التالية رمزيا علما بان المجموعة الكونية (الشاملة) هنا هي الاعداد الصحيحة الموجبة :

a) *if x is an even integer, then there is an integer y such that x =2y.*

أ) اذا كانت x عدد زوجي , فانه يوجد عدد صحيح y , ويكون $x = 2y$.

b) *There is a prime integer x such that x=2y for some y.*

ب) اذا كانت x عدد اولي ., فانه يوجد $x = 2y$ لبعض قيم y .

c) *For all prime integers x, if x>2, then x is not an even integer.*

ج) لكل الاعداد الاولية الصحيحة , اذا كانت x اكبر من 2 , فان x ليست عدد زوجياً.

الحل:

الفكرة من هذا المثال هنا , هي البحث عن عبارات تكون فيها $t(x,y), s(x)$, $p(x)$ و $r(x)$ بـدائل جيـدة , وكذلك البحث عن عبارات مثل *"there is, for each, for some"* ممكن ترجمتها بـ $\forall$ or $\exists$.

أ) - $a) - s(x) \to \exists y(t(x)) \quad or \quad s(x) \to \exists y \quad t(x,y)$

ب)- $b) - \exists x(p(x) \land \exists y(t(x,y)))$

ج)- ان الجملة الرمزية هنا لايمكن الحصول عليها لان المجموعة الكونية تمثل كل الاعداد الصحيحة وليست الاعداد الاولية الصحيحة , لذا يجب تغيير صياغة الجملة (*for all prime integers x,*) الى :

"for all integers x, if x is a prime, then x is not an even integer"

لان جملتنا حول الاعداد الصحيحة الاولية. اذن الجملة الرمزية ستكون $\forall x(p(x) \to (r(x) \to \neg s(x)))$.

▲

**

من جانب اخر هناك الرمز $\exists!$, فمثلا لو اخذنا الجملة $\exists! x$ فان ذلك يعني انه(توجد قيمة وحيدة لـ x مثل هذا........) , او انه توجد x وفقط x , الى حد ان ....):

" *there exists a unique x such that…* " *or there is one and only one x such that…* :

, فمثلا في الجملة $\exists! x[x < 1]$ ان تعيين قيمة الصفر لـ x يجعل $x < 1$ "" صحيحاً.

مثال 1-41

مستخدما المختصرات التالية :

1-اذا كانت $s(x)$ تقوم مقام الجملة : *x* عدد زوجي صحيح $"x \text{ is an even integer}"$

2- اذا كانت $t(x,y)$ تقوم مقام الجملة $x = 2y$.

3- اذا كانت $p(x)$ تقوم مقام الجملة : *x* هي عدد اولي صحيح $"x \text{ is prime integer}"$

4- اذا كانت $r(x)$ تقوم مقام الجملة $x > 2$

اعد كتابة الجمل التالية باللغة الانكليزية علما بان المجموعة الكونية هنا هي الاعداد الصحيحة الموجبة :

(a) $\forall x(p(x) \wedge \exists(t(x,y) \wedge r(x)))$

(b) $\neg\exists x(p(x) \wedge \exists y(t(x,y)) \vee r(z))$.

الحل:

(*a*) *for each integer x, x is a prime , and there is an integer y such that x is 2y and x is greater than 2.*

(*b*) *There is no integer x such that either both x is prime and there is a y such that x=2y or z is greater than 2.*

1-18 المتغير المُقيَّد وغير المُقيَّد Bound and Free variables

1- ان الـ $P(x,y,z)$ *"predicate"* ممثلاً بـ $(x+y=z)$ له ثلاثة متغيرات كل منهما غير مُقيَّد (free) في التعبير $P(x,y,z)$. لكن اذا عينا لـ x قيمة 2 $(x=2)$ فان النتيجة هي predicate بمتغير محدد هو $P(2,y,z)$. وهذا يكافئ الى *predicate* مكون من متغيرين غير مُقيَّدين مثل $Q(y,z)$ حيث $Q(y,z)$ هو صحيح اذا $(2+y=z)$. وبشكل مشابه ان $\exists y p(x,y,z)$ له متغيرين غير مُقيَّدين هما y و z .

2- اذا افترضنا الجملة التالية:

اذا كانت x عدد زوجي صحيح , فانه يوجد عدد صحيح y الى الحد انه x=2y .

if x is an even integer, then there is an integer y such that x = 2y

ويمكننا ان نعبر عن هذه الجملة بالرموز :

$s(x) \rightarrow \exists y(t(x,y))$ وفي هذه الجملة فقط ان المتغير y له مقدار غير محدد.

3- في التعبير $\forall x(p(x)....)$ او $\exists x(p(x)....)$ ان جزء التعبير الذي له نستخدم $\forall x$ او $\exists x$ يسمى المجال scope , ويسمى المتغير x محددا بواسطة $\forall x$ او $\exists x$ اذا وقع ضمن مجال $\exists x$ او $\forall x$. فالذي لايحدد بـ $\forall x$ او $\exists x$ يقال عنه free . اذن المجال scope لـ $\forall x$ او $\exists x$ هو جزء من التعبير فيه تكون المتغيرات مقيدة بهما.

مثال 1-42

في اي من الصيغ التالية , صف المتغيرات المُقيّدة (*bounded variables*) وغير المُقيّدة (*free variables*) .

(a) $\forall x(p(x) \wedge \exists y(t(x,y) \wedge r(x)))$

(b) $\neg\exists x(p(x) \wedge \exists y(t(x,y)) \vee r(z))$

(c) $\neg\exists x(p(x) \wedge \exists y(t(x,y)) \vee r(y))$

الحل:

في الجزء (a) ان المجال scope لـ $\forall x$ هو المتبقي من الصيغة , اي الذي يلي $\forall x$. اما مجال $\exists y$ فهو الصيغة $(t(x,y) \wedge r(x))$, وفي هذه الحالة لاتوجد متغيرات غير محددة (*free variables*) .

في الجزء (b) , ان مجال $\exists x$ هو الجزء المتبقي من الصيغة اي الذي يلي $\exists x$, ومجال $\exists y$ هو $t(x,y)$, ان المتغير z هو الغير محدد , لكن x و y هما المحددان (*bounded variables*).

في الجـزء (c) ان المجـال هـو نفسـه كـما في الجـزء (b) , الــ y في $r(y)$ هـو *free* , لكـن y في $t(x,y)$ محـددة.
▲

1-18-1 تعويض الثوابت للمتغيرات: *Substitution of constants for variables*

يمكننا تعويض عناصر المجموعة الكلية "*the universe*" للمتغير كما سنرى في المثال التالي:

مثال *43-1*

- افرض ان $s(x,y)$ تقوم مقام الجملة $x = 3y$. اعد كتابة الجمل التالية رمزياً:

(a) *there is an x such that* x = 3.4.

(b) *for all y*, 15 = 3.y.

(c) 15 = 3.5.

الحل:

(a) $\exists x(s(x,4))$

(b) $\forall y(s(15,y))$

(c) $s(15,5)$ ▲

19-1 قيم ومكافئ الجمل المُقدَّرة Truth and Equivalence of Quantified statements

افـرض ان $s(x)$ هـي جملـة حـول المتغـير x , فسـتكون الجملـة $\forall x(s(x))$ خاطئـة اذا كانـت $s(x)$ ليسـت صحيحة في المجوعة الكلية(الكونية) , وهذا يعني ان $\forall x(s(x))$ خاطئة اذا فقـط هنـاك قيمـة (*counterexample*) لـ $s(x)$.

مثال 1-44

1- اذا كانت $s(x)$ هي الجملة "" $x^2 \geq 0$ " و اذا كانت $t(x)$ هـي الجملة $x \geq 0$ "" وكلاهـما حـول الاعـداد الصحيحة . فهل $\forall x s(x)$ و $\forall x t(x)$ صحيحتان؟ وهل $\exists x(s(x))$ و $\exists t(s(x))$ صحيحتان؟

الحل:

بما ان $x^2 \geq 0$ هي دائما صحيحة لأي عدد صحيح لأنها مربعة , فان $\forall x s(x)$ هي صحيحة ايضاً. وبما ان $t(x)$ هي خاطئة للعدد الصحيح " $x = -1$ " فان $\forall x t(x)$ هو خاطئ.

2- بالنسبة للجملة $\exists x(s(x))$ فانها تكون صحيحة اذا $s(x)$ كانت صحيحة لقيمة واحدة لـ x على الاقل في المجموعة الكونية , وهذا يعني ان هناك حد واحد يجعل $s(x)$ صحيحة . كما ان $\exists t(s(x))$ هي خاطئة اذا $s(x)$ خاطئة لكل قيم x , هذا بمعنى ان المجموعة الحقيقية لـ $s(x)$ مجموعة فارغة.

▲

مثال 1-45

- اذا كانت $t(x)$ هي الجملة " " $x > 0$ حول الاعداد الصحيحة , فهل $\exists x(t(x))$ صحيحة؟ , و اذا كانت $s(x)$ هي الجملة "" $x^2 < 0$ فهل $\exists x(s(x))$ صحيحة؟

الحل: بما ان العدد 1 هو عدد صحيح لـ x , فان x اكبر من صفر $x > 0$, اذن $\exists x(t(x))$ صحيحة , وبما ان $x^2 < 0$ خاطئة لأي عدد صحيح x , اذن $\exists x(s(x))$ خاطئة.

▲

مثال 1- 46

اشرح لماذا الجملة $\neg\forall x(s(x))$ مكافئة للجملة $\exists x(\neg s(x))$. ؟

الحل :

ان الجملة $\forall x(s(x)$ هي صحيحة اذا وفقط اذا (*if and only if*), $s(x)$ هي صحيحة ولكل قيم x في الكون "*universal*" , وبالتالي فان $\neg\forall x(s(x))$ هي صحيحة اذا وفقط اذا (*if and only if*) توجد قيمة واحدة على الاقل لـ *x* بها تكون $s(x)$ خاطئة . والان $\exists x.(\neg s(x))$ **هي** صحيحة اذا وفقط اذا (*if and only if*), توجد قيمة واحدة على الاقل لـ x فيها $\neg s(x)$ صحيحة. وهلم جرا.

فان $\exists x.(\neg s(x))$ صحيحة اذا وفقط اذا (*if and only if*), توجد قيمة واحدة على الاقل لـ x فيها $s(x)$ خاطئة. هكذا وبالتالي لايهم اي مجموعة كونية نستخدم لـ $s(x)$, ان $\neg\forall x(s(x))$ و $\exists x(\neg s(x))$ سيكونان صحيحان لنفس الجمل $s(x)$ بالضبط. (بالضبط الظروف نفسها) . وبالتالي فان $\neg\forall x(s(x))$ و $\exists x(\neg s(x))$ متكافئتان لاي مجموعة كونية U وجملة $s(x)$. اذن هما متكافئان كتعبير مُقدَّر او تخميني *Quantified Expression* .

▲

وبنفس البرهان في هذا المثال يمكننا توضيح المتكافئات في النظرية التالية:

نظرية:

التعابيرالمقدرة) المقاسة("*quantified expressions*" التالية هي متكافئات "equivalent "

(a) $\forall x(\neg s(x)) \Leftrightarrow \neg\exists x(s(x))$

(b) $(\forall x \quad s(x)) \wedge t \Leftrightarrow \forall x(s(x) \wedge t)$

$$(\exists x \quad s(x)) \wedge t \Leftrightarrow \exists x(s(x) \wedge t)$$

(c) $\forall x \quad s(x)) \vee t \Leftrightarrow \forall x(s(x) \vee t)$

$$(\exists x\, s(x)) \vee t \Leftrightarrow \exists x(s(x) \vee t)$$

(d) $(\forall x p(x)) \wedge (\forall x q(x)) \Leftrightarrow \forall x(p(x) \wedge q(x))$

(e) $(\forall x p(x)) \vee (\forall x q(y)) \Leftrightarrow \forall x(p(x) \vee q(x))$

مثال 1-47

في الجملة التالية: اوجد الـ *predicates* , P الذي يجعل المعنى المتضمن *Implication* خاطئ.

$$\forall x \exists! y P(x,y) \Rightarrow \exists! y \forall x P(x,y)$$

ان P(x,y) يدل على x+y=0 , راجع جدول القيم الحقيقة (*truth table*) جدول (1-2) لمايدل على الضمن (p implies q) $p \Rightarrow q$ والجملة الشرطية كما اشرنا له في بدايات هذا الكتاب , فانه عندما يكون خاطئاُ فهذا يعني ان p هي صحيحة وان q هي خاطئة . لذا فان $\forall x \exists! y P(x,y)$ هو صحيح (True) وان $\exists! y \forall x P(x,y)$ هو خاطئ.

▲

ملاحظات:

1- لكل زوج من الاعداد الصحيحة x و y , فانه يوجد z كما في المعادلة $x + y = z$. ان هذه الجملة يمكن كتابتها بدلالة الصيغة التالية:

$$\forall x \forall y \exists z [x + z = y]$$

ان نفيها هو الجملة :

$$\neg \forall x \forall y \exists z [x + z = y]$$

وان مكافئها هو :

$$\neg \forall x \forall y \exists z [x + z = y] \Leftrightarrow \exists x \exists y \forall z \neg [x + z = y]$$
$$\Leftrightarrow \quad \exists x \exists y \forall z [x + z \neq y]$$

2- اما الجملة التالية $\neg \exists x \forall y \forall z P(x,y,z)$ فان مكافئها كمايلي:

$$\neg \exists x \forall y \forall z P(x,y,z) \Leftrightarrow \forall x \neg \forall y \forall z P(x,y,z)$$
$$\Leftrightarrow \forall x \exists y \neg \forall z P(x,y,z)$$
$$\Leftrightarrow \forall x \exists y \forall z \neg P(x,y,z)$$

3- ان التعبير $\neg \forall x P(x)$ هو خاطئ , وان التعبير $\exists x \neg P(x)$ يعني (لبعض قيم x , ان $P(x)$ هو خاطئ.

اي ان التعبيرين متكافئان: $\neg \forall x P(x) \Leftrightarrow \exists x \neg P(x)$.

4- ان التعبير $\neg \exists x P(x)$ يعني (انه من الخطأ ان توجد x الى حد ان p(x) هو صحيح.

" it is false that there exists an x such that p(x) is true "

وان التعبير $\forall x \neg P(x)$ يعني (لايوجد x الى الحد ان $P(x)$ هو صحيح , او لكل قيم x ان $P(x)$ هو خاطئ.) : *" there does not exist an x such that p(x) is true or for all x, p(x) is false"*

اي ان التعبيرين متكافئان: $\neg \exists x P(x) \Leftrightarrow \forall x \neg P(x)$

2- امثلة عن الجُمل المُقدَّرة والادوات المنطقية

<u>Quantifiers and logical operators Examples</u>

ان الجمل الرياضية " *mathematical statements* " غالبا ماتتضمن *quantifiers, predicates* , وادوات ربط منطقية "*logical operators*" مثل تلك التعابير بالامكان ان تاخذ صيغ مختلفة كما في الامثلة التالية :

افرض ان المجموعة الكلية الشاملة (*universe*) هي الاعداد الصحيحة وافرض ان:

1- N(x) تدل على الاعداد الصحيحة غير السالبة (*x is a none negative integer*) , وان E(x) هي ان x عدد زوجي (*x is even*) , و O(x) تدل على ان x عدد فردي (***x is odd***) , واخيرا *p*(x) تدل على ان x عدد اولي (***x is prime***) . فان الامثلة التالية تمثل نسخة التعابير بدلالة مجموعة الرموز المنطقية "*logical notation*".

أ) يوجد هناك عدد صحيح زوجي : There exists an even integer

$$\exists x E(x)$$

ب) كل عدد صحيح اما زوجي او فردي : ..Every integer is even or odd.

$$\forall x[E(x) \vee O(x)]$$

ج) جميع الاعداد الاولية الصحيحة ليست سالبة :

All prime integers are none negative

$$\forall x[(P(x) \Rightarrow X(x)]$$

د) العدد الزوجي الاولي فقط هو 2 : .The only even prime is two

$$\forall x[(E(x) \wedge P(x)) \Rightarrow x = 2]$$

ه) هناك عدد زوجي اولي واحد وفقط واحد:

There is one and only one even prime

$$\exists ! x[E(x) \wedge P(x)]$$

و) ليست كل الاعداد الصحيحة فردية : Not all integers are odd

$$\neg \forall x O(x), or \exists x \neg O(x)$$

ر) ليست كل الاعداد الاولية فردية: Not all primes are odd:

$$\neg \forall x[P(x) \Rightarrow O(x)], or \quad \forall x [P(x) \wedge \neg O(x)]$$

ز) اذا كان عدد صحيح ليس فردياً, فانه زوجي:

If an integer is not odd, then it's even.

$$\forall x[\neg O(x) \Rightarrow E(x)].$$

ملاحظة : الاداة $\Rightarrow$ تدل على وجود الشرط كما رأينا في الجمل الشرطية انفة الذكر .

2- اعتبر ان المجموعة الكلية تمثل الاعداد الصحيحة وافرض ان $P(x,y,z)$ يدل عـلى $xy = z$. فـان مـايلي هـو امثلـة عـلى الجمـل الرياضـية " *mathematical statements*" والصـيغ المتكافئـة *equivalent formulations* بمجموعـة الرمـوز المنطقية :

A) "*if x = 0, then xy=z for all values of y.*"

$$\forall x[\, x = 0 \Rightarrow \forall y P(\, x,y,z\,)]$$

B) "*if xy = x for every y, then x=0.*"

$$\forall x[\, \forall y P(\, x,y,z\,) \Rightarrow x = 0\,]$$

C) "*if xy ≠ z for some y, then x ≠0.*"

$$\forall x[\, \exists y \neg P(\, x,y,z\,) \Rightarrow \neg(\, x = 0\,)]$$

لاحظ ان التعبير في B لايمكن وصفه بـ $\forall x \forall y[\, P(\, x,y\,) \Rightarrow x = 0\,)]$ لان التعبير الاخير يمثل التعبير الخاطئ التالي:

"*for all x and y, if xy =x, then x=0.*"

ان القيمة (1) لـ x والقيمة (1) لـ y تمثل قيم جعل التعبير خاطئ اي انها counterexamples بيـنما التعبـير في B هـو صحيح .

مثال 1-48

اكتب الجمل التالية كجمل بشكل صبغ *formulated statements* :

1-كل الاعداد الصحيحة اكبر من 10 : *All integers greater than 10*:

$$\forall x[x > 10]$$

2-المجموعة الكلية تحتوي على ثلاثة فقط . *The universe contains only 3*

$$\forall x[x = 3]$$

3-لكل قيم x , ان $p(x)$ هو صحيح. *For all x, p(x) is false*

$$\forall x \neg p(x)$$

▲

مثال 1-49 اذا كانت :

1- $S(x,y,z)$ تدل على الـ **predicate** $(x+y=z)$

2- $P(x,y,z)$ تدل على الـ **predicate** $(x.y=z)$

3- $L(x,y)$ تدل على الـ **predicate** $(x<y)$

افرض ان المجموعة الكلية هي مجموعة الاعداد الطبيعية (*natural numbers*) اي : $N\{1,2.....)$, اكتب الجمل التالية بدلالة الرموز *formulated statements* :

1- x ليست اقل من صفر : *No x is less than 0*

$$\neg\exists x[L(x,0)] \text{ او } \forall x[\neg L(x,0)]$$

2- لكل **x** و **y** , ان هناك **z** الى الحد ان $x+y=z$:

For every x and y, there is a z such that x + y = z

$$\forall x\forall y\exists zS(x,y,z)$$

▲

1-20 البرهان : *The proof*

ان النظام الرياضي (*mathematical system*) يشتمل على بديهيات او حقائق مقررة (*axioms*) , وتعريفات (*definitions*) , ومصطلحات غير معرفة (*undefined terms*) . فالبديهيات يفترض انها صحيحة *true* . اما التعريفات فتستخدم لابتكار مفاهيم جديدة بالاستناد الى المفاهيم الصحيحة الموجودة قبلها. وان بعض المصطلحات من الصعب تعريفها بسهولة لكن ذلك ممكن بواسطة البديهيات . كما انه يمكننا اشتقاق بديهيات في النظام الرياضي الذي نتحدث عنه هنا.

اما النظرية (*theorem*) فهي قضية (*proposition*) قد بُرهنت لتكون صحيحة .وان بعض انواع النظريات يشار لها كاستدلالات او نظريات اشتقت من نظريات اخرى (*corollaries*),أو الى نظريات غيرمهتمين بصحتها (*lemmas*) لكن تستخدم لبرهنة نظريات اخرى .

ان الحُجَّة التي تقيم الصدق للنظرية تسمى البرهان (***proof***).اما المنطق (*logic*) فهو الاداة لتحليل البراهين . وفي هذا الجزء سنصف بعض الطرق العامة في البرهان وسنستخدم المنطق لتحليل الحُجج الصحيحة والباطلة (*valid and not invalid arguments*) .وكذلك سندرس الاستنتاج (الاستقراء) الرياضي (*mathematical induction*) لاحقاً, و براهين الحل (*resolution proofs*) اللذين هما طرق برهان خاصة .وفيمايلي سنتناول بعض الامثلة التي تخص الانظمة الرياضية (*mathematical systems*).

مثال 1-50

1- الهندسة الاقليدية (*Euclidean geometry*) تزود مثالا في النظام الرياضي . **فمن بين البديهيات** " *axioms* " مايلي:

1- اعطاء نقطتين (*two points*) محددتين فانه يوجد بالضبط خط مستقيم واحد (*line*) يشتمل عليهما.

2- اعطاء خط line ونقطة منفصلة عنه (*distinct point*) فانه يوجد خط واحد بالضبط يوازي الخط المار بالنقطة .

ان مصطلحات خط (*line*) و نقطة (*point*) هي مصطلحات غير معرفة , والتي هي سهلة التعريف باستخدام البديهيات التي تصف خواصهما.

تعريف 1-10 ومن بين التعريفات (*definitions*)

1- يتطابق المثلثان اذا كان بالامكان تطابق رؤوسهما لكي تكون اضلاعهما وزواياهما المتماثلة متساوية

2- تكون الزاويتان متكاملتان (*Supplementary*) اذا كان مجموع قياسهما **180** درجة.

مثال 1-51

الاعداد الحقيقية تزودنا بمثال اخر عن النظام الرياضي " *mathematical system* "

فمن بين البديهيات :

1- لكل الاعداد الحقيقية ***x*** و ***y*** , ان $xy = yx$ وهذه سهل برهانها.

For all real numbers x and y , xy = yx.

2- هناك مجموعة جزئية **P** للاعداد الحقيقية تحقق (أ) و (ب) ادناه:

there is a subset P of real numbers satisfying:

أ) اذا كانت ***x*** و ***y*** في **P** , فان $x + y$ و xy ايضا في **P**

*If x and y are in **P** , then x+y and xy are in **P** .*

ب) اذا كانت **x** عدد حقيقي , فانه بالضبط واحد من الجمل التالية هو صحيح:

If x is a real number, then exactly one of the following statements is true.

$$x \text{ is in } P, \qquad x = 0, \qquad -x \text{ is in } P$$

تعريف 1-11

1- ان العناصر في **P** المشار اليها في البديهيات اعلاه تدعى اعداد حقيقية موجبة .

2- القيمة المطلقة ($|x|$) لـ x , تعرف لتكون x اذا x موجبة او صفر و $-x$ ماعدا ذلك.

وفيمايلي نعطي بعض الامثلة للنظريات (*theorems*) والاستدلالات النظرية (*corollaries*) و مايسمى بـ (*lemmas*) المشار اليهما في فقرة البرهان اعلاه .وذلك من خلال الهندسة الاقليدية ونظام الاعداد الحقيقية .

مثال 1-52 للنظريات في الهندسة الاقليدية :

1- اذا كان ضلعان في مثلث متساويين فان الزوايا المقابلة لهما متساويتان.

if two sides of a triangle are equal, then the angles opposite them are equal.

2- اذا كان قطري شكل رباعي ينصف احداهما الاخر فان الشكل الرباعي هو متوازي اضلاع.

If the diagonals of a quadrilateral bisect each other, then the quadrilateral is a parallelogram.

مثال 1-53 على الاستدلالات

1- اذا كان مثلث متساوي الاضلاع فانه متساوي الزوايا.

If a triangle is equilateral, then it is equiangular.

وهذه مشتقة حالاً من النظرية الاولى في المثال (1-52) اعلاه.

مثال 1-54 على النظريات حول الاعداد الحقيقية :

1- ان x مضروبة في صفر تساوي صفراً ($x.0 = 0$) لكل عدد حقيقي x.

$x.0 = 0$ *for every real number* x .

2- لكل الاعداد الحقيقة x, y و z , اذا كانت x اقل او تساوي y , و y اقل او تساوي z , فان x اقل او تساوي z .

For all real numbers x, y and z, if $x \le y$ *and* $y \le z$ *, then* $x \le z$

مثال 1-55 عن النظرية التي تستخدم لبرهان نظرية اخرى ـ lemmas حول الاعداد الحقيقية :

1- اذا كان n عدد صحيح موجب , فانه اما $n-1$ هو موجب او $n-1=0$

if n is a positive integer, then either n-1 is a positive integer or n-1=0.

بالتاكيد ان هذه النتيجة غير مهتمين بصحتها لكنها ممكن ان نستخدمها لبرهان نتائج اخرى:

من ناحية اخرى ان النظريات غالباً مايكون لها الصيغة التالية:

$$For\ all\ x_1, x_2, \ldots\ldots\ldots\ldots, x_n, if\ \ p(x_1, x_2, \ldots, x_n), then\ q(x_1, x_2, \ldots, x_n).$$

ان هذه الجملة الغير محددة الشاملة تزودنا بالجملة الشرطية التالية:

$$if\ \ p(x_1, x_2, \ldots x_n), then\ q(x_1, x_2, \ldots, x_n)$$

والتي هي صحيحة لكل القيم $(x_1, x_2, \ldots\ldots\ldots\ldots, x_n)$ في المجال الذي نتحدث عنه. ولبرهنة هذه الجملة الشرطية , نفترض ان $x_1, x_2, \ldots\ldots\ldots\ldots, x_n$ هي اعداد عشوائية في المجال الذي نتحدث عنه. فاذا كانت $p(x_1, x_2, \ldots, x_n)$ خاطئة , فانه طبقاً **للتعريف رقم 1-2 (وجدول القيم للجمل الشرطية1-2**), فان الجملة الشرطية المذكورة هي صحيحة. لهذا نحتاج الى الحالة التي فيها $p(x_1, x_2, \ldots, x_n)$ صحيحة. ان البرهان المباشر *a* (*direct proof*) يفترض ان $p(x_1, x_2, \ldots, x_n)$ صحيحة ومن ثم تستخدم $p(x_1, x_2, \ldots, x_n)$ بالاضافة الى نظريات اخرى ,والى تعاريف و نظريات مشتقة سابقا لبيان بشكل مباشر ان $q(x_1, x_2, \ldots, x_n)$ هي صحيحة.

مثال 1-56

اعطي بُرهاناً مُباشراً للجملة التالية:

لكل الاعداد الحقيقية d, d_1, d_2 و x انه:

اذا كانت d تساوي الاقل لـ $min\{d_1,d_2\}$ و x اقل او تساوي d , فان x اقل اوتساوي d_1 و d_2

if d= min{d_1,d_2} and $x \le d$ ***, then*** $x \le d_1$ ***and*** $x \le d_2$

البرهان:

اننا نفرض ان d, d_1, d_2 و x اعداد حقيقية عشوائية (*arbitrary real numbers*) وبذلك فانه يكفي ان نفرض ان:

$$d = min\{d_1, d_2\} \text{ and } x \leq d ,$$

هو صحيحاً . وم ثم نبرهن ان:

$$x \leq d_1 \text{ and } x \leq d_2$$

هو صحيح ايضاً.

من تعريف الاقل (*min*) فانه ينتج ان :

$$d \leq d_1 \text{ and } d \leq d_2$$

ومن العلاقة:

$$x \leq d \text{ and } d \leq d_1$$

فاننا نستنتج ان $x \leq d_1$ (انظر النظرية السابقة الجزء الثاني **في المثال 1-54**)

ومن : $x \leq d$ and $d \leq d_2$ فاننا نستنتج ان $x \leq d_2$ بالاستناد الى نفس النظرية في المثال 1-54 ,

لذا فان :

$$x \leq d_1 \text{ and } x \leq d_2$$

▲

من جانب اخر فانه توجد هنـاك تقنيـة ثانيـة للبرهـان تـدعى البرهـان بـالنقض (*proof by contradiction*) . ان البرهـان بالنقض **يؤسس المثال 1-55 اعلاه** وذلك بفرض ان الفرضية p هي صحيحة وان النتيجة q هي خاطئة ومن ثم اسـتخدام p و $\neg q$ بالاضافة الى بديهيات اخرى وتعاريف ونظريات مشـتقة نسـتنتج الـنقض للبرهـان (*contradiction*) . ان البرهـان بالنقض هو قضية (*proposition*) بالصيغة التالية:

$$(r \wedge \neg r)$$

, حيث r هو ايُّ عبارة مهما تكون. ان البرهان بالنقض يدعى احياناً البرهان غير المباشر (*indirect proof*) نظـرا لتأسـيس المثال 1-55 مستخدمين البرهان بالنقض فانه يمكن لأحد ان يستنتج ($r \wedge \neg r$) ومن ثم يستنتج ان العلاقـة في المثـال 1-55 صحيحة.

ان الفرق اللوحيد بين الفرضيات في البرهان المباشر والبرهان بالنقض هو النتيجة المنفية (negated conclusion) , حيـث في البرهان المباشر ان النتيجة المنفيةلا تفترض

(*not assumed*) لكنها تفترض(*is assumed*) في البرهان بـالنقض. ان البرهـان بـالنقض ربمـا يُسـوِّغ بملاحظـة ان العبارتان :

$$p \to q \qquad and\ p \wedge \overline{p} \to r \wedge \overline{r}$$

متكافئتان , **انظر الجدول(20-1) التالي**:, حيث $\overline{p}$ هي نفسها $\neg p$ ونفس الشئ بالنسبة لـ $\overline{r}$.

p	**q**	**r**	$p \to q$	$p \wedge \neg q$	$r \wedge \overline{r}$	$p \wedge \overline{q} \to r \wedge \overline{r}$
T	T	T	T	F	F	T
T	T	F	T	F	F	T
T	F	T	F	T	F	F
T	F	F	F	T	F	F
F	T	T	T	F	F	T
F	T	F	T	F	F	T
F	F	T	T	F	F	T
F	F	F	T	F	F	T

جدول 1-20

مثال 1-57

في هذا المثال سنعطي برهاناً بالنقض (*proof by contradiction*) للجملة التالية:

لكل الاعداد الحقيقية x و y , اذا كانت $x + y \geq 2$ فانه اما $x \geq 1$ او $y \geq 1$

For all real numbers x and y, if $x + y \geq 2$ *, then either* $x \geq 1$ *or* $y \geq 1$

البرهان:

افرض ان النتيجة (*conclusion*) ($y \geq 1$ *or* $x \geq 1$) هي خاطئة حسب تعريف البرهان بـالنقض . اذن $x < 1$ و $y < 1$ (تذكر ان نفي الـ OR's ينتج الـ ANDs). انظر ايضـا الى قـوانين ديموركـان للمنطـق (***De Morgan laws for logic***) في **المثال**1-16 .

استخدام النظرية السابقة , نضيف المتباينات (*inequalities*) $x < 1$ و $y < 1$ لنحصل على :

$$x + y < 1 + 1 = 2$$

وهذه تمثل $\neg p$.

عند هذه النقطة تم اشتقاق النقض $p \wedge \neg p$, حيث في الجملة الاصلية ان :

$$p : x + y \geq 2$$

وهو يناقض ماحصلنا عليه (اي) $\neg p$. ومن هنا نستنتج ان الجملة في مثالنا هذا صحيحة.

من جانب اخر افرض اننا اعطينا البرهان بالنقض للعلاقة في المثال 1-55 والتي فيها كما هو في هذا المثال, اننا استدللنا (*deduced*) على $\neg p$. وبالواقع اننا برهنا :

$$\overline{q} \rightarrow \overline{p}$$

وهذه الحالة الخاصة من البرهان بالنقض تسمى البرهان ضد الحقيقة (*contra positive*) اي عكس النتيجة الفعلية الحاصلة . ان في بناء البرهان يجب ان نتاكد ان الحجج المستخدمة صحيحة (*valid*).

▲

مثال 1-58

الحجة (*an argument*) هي تتابع من العبارات تكتب كمايلي:

$$\begin{array}{l} p_1 \\ p_2 \\ \vdots \\ p_n \\ \hline \therefore q \end{array}$$

Or $\quad p_1 , p_2 ,......p_n / \therefore q$

ان العبارات $p_1 , p_2 ,........p_n$ تدعى (*hypothesis (or premises)*) والعبارة q تدعى النتيجة *conclusion* .ان الحجة الصحيحة تزودنا بمايلي: انه اذا كانت p_1 , p_2 and p_n كلها صحيحة فان q يجب ان تكون صحيحة. وغير ذلك ان الحُجَّة غير صحيحة(*invalid*).

▲

اما في الحجة الصحيحة (*true argument*) , اننا احيانا نقول ان النتيجة تستنتج من الفرضية . لاحظ اننا لم نقل ان النتيجة صحيحة true . نحن قلنا فقط اذا انت اخذت الفرضية كأمر مسلم به, فانت يجب ان تأخذ النتيجة كأمر مسلم به ايضاُ. اخيراً ان الفرضية صحيحة بسبب صيغتها وليست بسبب محتواها.

مثال 1-59

حدد فيما اذا كانت الحجة *argument* التالية صحيحة :

$$p \to q$$
$$p$$
$$\overline{\qquad\quad}$$
$$\therefore q$$

الحل الاول:

ننشئ جدول القيم لكل العبارات المكونة للحجة *argument* وكمايلي:

ومن خلال جدول القيم اننا نرصد انه عندما $p \to q$ و p كلاهما صحيح فان النتيجة q صحيحة ايضاُ انظر جدول (2-1) التالي الذي سبق وان ذكرناه فيبداية هذا الفصل, وبذلك فالحجة صحيحة (*argument is valid*).

p	q	$p \to q$	p	q
T	T	T	T	T
T	F	F	T	F
F	T	T	F	T
F	F	T	F	F

والحل الثاني :

هو انه بالامكان تجنب كتابة جدول القيم وبالاثبات المباشر هو انه مادامت النظرية صحيحة فان النتيجة ايضاُ ستكون صحيحة. افرض ان $p \to q$ و p كلاهما صحيح , ومن ثم فان q يجب ان تكون صحيحة وفيماعدا ذلك فان $p \to q$ يجب ان تكون خاطئة . لذلك فان الحجة صحيحة.
▲

1-20-1 الاستنتاج المضاد للواقع او للحقيقة *Contrapositive Inference* :

ان الصيغة المضادة للحقيقة (*contrapositive form*) للجمل الشرطية تقودنا الى مبدأ الاستنتاج المضاد للحقيقية (*contrapositive inference*) وهي كمايلي:

From $\neg q \rightarrow \neg p$ **we may conclude** $p \rightarrow q$.

اي من **الجملة الشرطية** $\neg q \rightarrow \neg p$ نستنتج **الجملة الشرطية** $p \rightarrow q$

مثال 1-60

استخدم البرهان بالاستنتاج المضاد للحقيقة لبرهنة انه لكل عنصر n في الكون للاعداد الصحيحة الموجبة انه اذا كان مربع n اكبر من مئة ($n^2 > 100$) , فان n اكبر من 10 .

$$if \ \ n^2 > 100, then \ \ n > 10.$$

البرهان:

1- سوف نفترض الجملة $\neg(n > 10)$ اي ان n ليست اكبر من 10 , ونشتق الجملة $\neg(n^2 > 100)$ موضحين ان $\neg(n > 10)$ تشتمل ضمناً $\neg(n^2 > 100)$ اي ($\neg(n > 10) \ \ implies \ \ \neg(n^2 > 100)$), وبالتالي سوف نبين ان :

$$\neg(n > 10) \ \ \rightarrow \ \ \neg(n^2 > 100)$$

هي صحيحة لكي بالاستنتاج المضاد للنتيجة الفعلية نبين ان :

$$(n^2 > 100) \rightarrow (n > 10)$$

2- نفترض ان **n** ليست اكبر من 10 ($\neg(n > 10)$) وهذا يعني ان n اقل او تساوي عشرة ($n \leq 10$) ومن ثم فان :

$$n \,.\, n = n.n \leq 10.10$$

وهذا يمعنى انه اذا كانت n ليست اكبر من 10 (اقل من 10)., فان مربعها n^2 ايضا ليس اكبر من 100 (اقل من 100) .

اذن وبواسطة الاستنتاج المضاد للحقيقة , اي العكس صحيح ايضا وهو اذا $n^2 > 100$ فان $n > 10$:

$$if \ \ n^2 > 100, then \ \ n > 10.$$

وهذا حسب الصيغة التي اشرنا اليها في بداية الاستنتاج المضاد للنتيجة الفعلية وهي :

from $\neg q \to \neg p$ **we may conclude** $p \Rightarrow q$.

مثال 1-61

استخدم البرهان بالنقض(*proof by contradiction*) لبيان ان $\sqrt{7}$ عدد غير نسبي(*irrational number*).

الحل:

1-نفرض ان $\frac{n}{m} = \sqrt{7}$ وباقل نسبة عامل مشترك (*lowest terms*) اي 1 , اذن $m\sqrt{7} = n$

(ضرب الطرفين بالوسطين) ومن ذلك يكون: $7m^2 = n^2$.

2- وبما ان العدد 7 هو عدد اولي ولأجل ان يكون معاملا لـ n^2 , فانه يجب ان يكون معاملاً لـ ***n*** . لكن ان تكون الـ 7 عاملاً لـ ***n*** , فان ***n*** يجب ان تكون **7k** لبعض القيم الصحيحة **k** . ومن ذلك يكون:

$$7m^2 = (7k)^2 = 49k^2$$

اذن:

$$m^2 = 7k^2$$

لكن وكما في ***n*** فان 7 يجب ان تكون عاملاً ايضاً لـ **m** . وبذلك تكون 7 عامل مشترك بين **m** و **n** , وهذا يناقض الفرضية في البداية (اقل نسبة عامل مشترك). وبسبب ان اي كسر عددي يعبر عنه باقل نسبة عامل مشترك . اذن الفرضية $\frac{n}{m} = \sqrt{7}$ فرضية خاطئة . لذلك $\sqrt{7}$ عدد غير نسبي. . ويمكننا بنفس الطريقة اثبات ان $\sqrt{3}$ و $\sqrt[3]{3}$ هما ايضا ليسا اعداد نسبية . ▲

1-20-2 البرهان المباشر ***Direct Proof*** **:**

ان النظرية الرياضية هي عبارة عن جملة لها الصيغة :

$$"\,if \quad p, q, \ldots, then \;\; t\,"$$

او اكثر وبشكل رمزي :

$$(p \wedge q \wedge .... \rightarrow t)$$

ان الجمل p و q تدعى بـ الفرضية (*hypotheses*) والجملة t تدعى النتيجة (*conclusion*), ولبيان مثل تلك الجملة صحيحاً true . فاننا نقيم حجة (argument) لبيان انه حينما تكون الفرضية صحيحة فان النتيجة صحيحة

وهذا مايفترضه البرهان المباشر (*direct proof*) .

البرهان المباشر للنتيجة من الفرضية هو سلسلة من الجمل :

$$\begin{array}{l} first \quad statement \\ second \quad statement \\ \vdots \\ last \quad statement \quad which \quad represent \quad the \quad conclusion \end{array}$$

ان الجملة الاخيرة تمثل النتيجة (*conclusion*) وان كل من الجمل الاخرى هي اما تمثل فرضية او تمثل حقائق او معلومات رياضية (*mathematical facts*) , او تمثل النتيجة لتطبيق الاستنتاج المباشر الى الجملتين السابقتين قبلها.

مثال 1-62

برهن باستخدم البرهان المباشر (*direct proof*) انه اذا العددين الصحيحين ***m*** و ***n*** من مضاعفات العدد **3** فان مجموعهما ($m+n$) هو مضاعفات **3**.

الحل:

1- **الفرضية:**

افرض ان كل من **m** و **n** هو من مضاعفات العدد **3**.

2- **الحقيقة الرياضية المقبولة :**

اذا كانت **m** من مضاعفات العدد **3** ,,, فانه توجد بعض الاعداد الصحيحة ***i*** الى الحد ان $m = 3i$.

3 - **الاستنتاج المباشر**

ان : $m = 3i$

نكرر الخطوات نفسها لـ **n** لنحصل في النتيجة على $n = 3j$.

اذن اذا $m = 3i$ و $n = 3j$, فان $m + n = 3i + 3j$

ولذلك الغرض فان :

$$m+n=3i+3j$$

وباخراج عامل مشترك بينهما فان:

$$m+n=3(i+j)$$

وهي حقيقة مقبولة . وبالتعريف اذا كان يوجد عدد صحيح k , حيث $m+n=3k$, فان $m+n$ تعتبر مضاعفاً لـ **3** . ومن اجل ذلك الغرض فان ($m+n$) هو مضاعفات **3** . وهذا الاخير يمثل استنتاجاً مباشراً

▲

مثال 1-63

انشئ برهاناً مباشراً انه اذا كانت **m** و **n** اعداد زوجية (*even numbers*) فان مجموعهما ايضا عدد زوجي . ملاحظة: (**n** تعني عدد زوجي اذا كان *n=2j* لبعض الاعداد الصحيحة *j*).

اذا كان n عدد زوجي , فحسب الملاحظة ان n=2j وهذه حقيقة مقبولة رياضياً تتعلق بتعريف العدد الزوجي.وبما ان **m** و **n** هما عددان زوجيان (الفرضية) فانه توجد بعض الاعداد الصحيحة ***i*** و ***j*** بحيث ان **m=2j** و **n=2i** . وهذا يعطينا ان :

$$m.n=2i.2j=2.(2ij)=2k,$$

حيث $2ij=k$. ومن ناحية ثانية " اذا كانت $mn=2k$ فان ***mn*** عدد زوجي", فان هذا يعني ان ***mn*** عدد زوجي (استنتاج مباشر).

▲

مثال 1-64

اكتب جدول القيم للجملة $p\leftrightarrow q$ والجملة $\neg(p\oplus q)$. والان اعطي برهاناً مباشراً (*direct proof*) انه اذا كانت **p** و **q** هما جملاً, فان $p\leftrightarrow q$ تكافئ الجملة $\neg(p\oplus q)$.انظر جدول (1-21)

p	q	p	$\leftrightarrow$	q	$\neg$	p	$\oplus$	q
T	T	T	T	T	T	T	F	T
T	F	T	F	F	F	T	T	F
F	T	F	F	T	F	F	T	T
F	F	F	T	F	T	F	F	F
step		1	2	1	3	1	2	1

جدول (1-21)

اذا $p \leftrightarrow q$ و $\neg(p \oplus q)$ كان لهما نفس الصف العمودي النهائي في جدول القيم لهما جدول (21-1) , فانهما متكافئان وهذه حقيقة مقبولة . ومع ذلك نرى ان لهما نفس الصف العمودي اعلاه بواسطة جدول القيم (الجزء المظلل بالجدول) , اذن هما متكافئان (استنتاج مباشر).

▲

مثال 1-65

انشئ برهاناً انه اذا كانت **m** و **n** كلاهما فردياً (*odd numbers*), فان مجموعهما عدداً زوجياً *even number* .

ملاحظة : تعريف العدد الفردي انه **n** تعتبر عدد فردي اذا كان $n = 2j + 1$ لبعض القيم الصحيحة ***j***.

n is ***odd if and only if*** *n = 2j + 1 for some j*

الحل :

اذا كانت **m** عدد فردي , فانه من التعريف ان $m = 2i + 1$ لبعض قيم **i** الصحيحة , واذا كان **n** عدد فردي ايضاً , فان $n = 2j + 1$ لبعض قيم **j** الصحيحة .

وعند جمع العددين الفردين **m** و **n** ينتج مايلي:

$$m + n = 2i + 2j + 2 = 2(i + j + 1)$$

وم ذلك اذا كانت $m + n = 2k$ لبعض القيم الصحيحة **k** , فان هذا يعني ان $m + n$ هو عدد زوجي. فبما ان $i + j + 1$ هو عدد صحيح لأن **i** و **j** و **1** كلها اعداد صحيحة . فاننا نستنتج ان $m + n$ هو عدد زوجي . ومن اجل ذلك اذا كانت **m** و **n** كلاهما فردياً (*odd numbers*) , فان مجموعهما عدداً زوجياً (*even number*).

▲

تمارين الفصل الاول

1-Convert the following statements into symbolic form, using *p* for $x < -1$ and *q* for $x^2 > 2$:

a- $x^2 > 2$ or $x < -1$, but not both. b- $x < -1$ or $x^2 > 2$

c- $x \geq -1$ and $x^2 > 2$ d- $x^2 \leq 2$

2- Show the logic diagram and discover the final outputs for all possible inputs for the following statements :

a- $p \vee (q \wedge r)$ b- $(p \wedge q) \vee (p \wedge r)$

3-Write down and explain the truth table for the symbolic compound statement given: $p \oplus (p \vee q)$

4-Draw the Venn diagram that illustrates the equivalence of the following compound statements:

$$p \vee (q \wedge r) \Leftrightarrow (p \vee q) \wedge (p \vee r)$$

5-Write down and explain the truth table for the symbolic compound statement given: $(p \oplus q) \wedge \neg (r \oplus q)$

6- Show the logic diagram and discover the final out put for all possible input for the following statement:

a- $(p \wedge (q \vee \neg p)$ b- $\neg p \vee \neg q$

7-Evaluate each proposition in the following for the truth values:

p= F, q = T, r = F

1- $\overline{P} \vee \overline{(q \wedge r)}$

2- $\overline{(P \vee q)} \wedge (\overline{p} \vee r)$

8-Write the truth table of the following:

1- $(p \vee q) \wedge \overline{p}$

2- $(\overline{p} \vee \overline{q}) \vee p$

9-Determine whether each statement is true or false

1- $5 < 7 \quad and \quad 9 < 7$

2-it is not the case that ($5 < 7 \quad and \quad 9 < 7$)

3-for every real number x, $x^2 \geq 0$

10- Write the statement: "if $2 < 5$, then $6 > 8$ " symbolically , and write the contrapositive and converse both symbolically and in words.Then find the truth value of each statement.

11- Show that $p \wedge q$ **implies** $p \vee (\neg p \wedge q)$.

12-Assuming that p is true , q is false, and r is true , find the truth value of each proposition:

$(p \wedge q) \rightarrow r$ 1

2- $(p \vee q) \to \bar{r}$

$p \wedge (q \to r)$ 3-

$p \to (q \to r)$ 4-

13-Verify the following proposition

$\overline{p \vee q} \equiv \bar{p} \wedge \bar{q}$

14-Write symbolically and find the truth value of each conditional statement and its converse

1- $if \quad 1 < 2 \text{ , } then \quad 3 < 6$

2- $if \quad 1 > 2 \text{ , } then \quad 3 < 6$

15- Prove each of the following statements :

1- $p \leftrightarrow q \equiv (p \to q) \wedge (q \to p)$

16-for every real number x, $if \quad x > 1, then \quad x + 1 > 1$ is true.

1-Assuming that *p* is true, *q* is false, and *r* is true, find the truth value of each statement:

a) $(p \vee q) \to \neg r$ b) $p \to (q \to r)$

c) When does the statement " if $x > 1$, then $x + 1 > 1$ " is true?

17- if P is proposition function prove each following proposition has the same truth value.

$$(\overline{\forall xP(x)}\;;\exists x\;\overline{p(x)})$$

18- Let *P*(*x*, *y*) be the statement:

$$if \quad x^2 < y^2, then \quad x < y$$

The domain of discourse is the set of real numbers, prove the statement:

for every x, every y, P(x, y) is false

19-using proof by contradiction to prove the following statement:

For all real numbers x and y, if $x + y \geq 2$ *, then either* $x \geq 1$ *or* $y \geq 1$

20-determine if the following argument is valid:

$$\begin{array}{l} p \to q \\ p \\ \hline \therefore q \end{array}$$

21- Prove the following using resolution:

1. $a \vee b$
2. $\overline{a} \vee c$
3. $\overline{c} \vee d$

$- - - - - - - -$

$\therefore b \vee d$

22- Find a counter-example to each of the following statements about positive integers n .

1- $2n^2 + n = n^3 - n^2 + 3n$

2- If n is a prime, then 12 and $n^3 - n^2 + n$ have a common factor.

23-Rewrite the sum $\sum_{i=0}^{n} ir^{n-1}$ replacing the index *i* by *j* , where *j*=*i*+1

24-If $\{a_i\}_{i=m}^{n}$ is a sequence, define the sigma and product notations of that sequence.

25 Is the statement $(3+5=35) \leftrightarrow (2+7=10)$ true or false? How?.

26- Find the truth table of the statement $(p \vee q) \wedge \neg(p \wedge q)$.

27- Show that for every real number x, the statement:

If $x > 1$, then $x+1 > 1$ is true. Then find the universally quantified statement, and explain if it is true or false.

28- Define the sequence $< t_n >$ by the rule $t_n = n^2 - 1, \quad n \geq 1$.

29- Let R be the relation on $X = \{1, 2, 3, 4\}$ defined by $(x, y) \in R$ if $x \leq y$,
$x, y \in X$. Find the relation R.

30-Given $g = \{(1,a),(2,b),(3,c)\}$ **a function from** $X = \{1, \; 2, \; 3\}$ **to** $Y = \{a,b,c\}$ **and** $f = \{(a,y),(b,x),(c,z)\}$ **a function from Y to** $Z = \{x,y,z\}$**.Find the composition function from** X **to Z.**

31- If the universe is the set of integer I, Find the predicate of each of the following statements, Then show the truth of each one:

$1 - \forall x[x < x+2]$

$2 - \forall x[x = 2]$

$3 - \forall x \forall y[x + y > x]$

32-Rewrite the following statement into logical notation:

a- if an integer is not odd, then it is even.

b- if $xy \neq z$ for some y, then $x \neq 0$

c- every integer is even or odd.

33- The symbolic statements $\neg(p \vee q)$ *and* $(\neg p \wedge \neg q)$ **are equivalent. Substitute the statement** $x \leq 0$ **for** ***p*** **and** $x \geq 10$ **for** ***q*****, using the integers as a universe. Find the truth sets of the resulting statements. Are the resulting statements equivalent?**

34- Draw a Venn diagram that illustrates the equivalence:

$$p \vee (q \vee r) \Leftrightarrow (p \vee q) \vee r$$

35-Using a proof by contrapositive inference, prove that, for each member n of the universe of positive integers , if $n^2 > 25$ **then** $n > 5$.

36- Give an example that explains why $\neg\exists x(p(x))$ is not equivalent to $\exists x(\neg p(x))$.

37- Prove that if p and *q* are statements, then ($p \rightarrow q$) is equivalent to ($\neg p \vee q$).

Hint ; use the truth table.

38- If the universe is the set of integers I, explain the following statement and determine when it is false and when it is true. Also; find the predicate of this statement:

$$\forall x \forall y [x + y > x]$$

39-Construct an indirect proof that if m^2 is even ,then m is even.

40- If p, r, and s are the statements; " x is prime integer", "x> 2" , and " x is even integer" respectively. Which of the following are true and which are false? . Give an example.

$(a)\forall x p(x)$ $\quad$ $(c)\exists x(p(x) \wedge s(x))$

$(b)\exists x p(x)$ $\quad$ $(d)\exists x(p(x) \wedge s(x) \wedge r(x)$

41- Rewrite the following statement in symbolic form, using the abbreviations s(x) for "x is an even integer". t(x,y) for" $x \leq y$, " r(x) for "x is a square," and *q*(*x* ,*y*) for " $y = x^2$, the universe for both x and y is the set of positive integers.

(a) For each y, if there is an x such that $y = x^2$, then y is square.

(b) For each x , if there is an x such that $y = x^2$, then x is square.

(c) y is square if there is an x such that $y = x^2$,

(d) For each x and y, if $y = x^2$, then $x \leq y$,

(e) There is no x such that x is a square and x is an even integer.

42-Write down the set of order pairs of the relation described the relationship between x and y given by x is related to y if $|x-y| \leq 1$ **on {1, 2, 3, 4}. And draw a digraph of this relation.**

الفصل الثاني

الاستنتاج (الاستقراء) الرياضي

MATHEMATICAL INDUCTION

1-2 المقدمة

في واحد من النماذج المثالية للحاسوب المتوازي (*parallel computer*) انه توجد عدة معالجات (*processors*) من ضمنها معالج اساسي (*master processor*) يتحكم في النظام بشكل كامل. وفي هذا النموذج كل معالج يرسل رسالة الى اثنين اخرين (ماعدا المعالج الاساسي) , وكل واحد يستلم رسالة من معالج واحد بالضبط. وعندما المعالج الاساسي لديه رسالة يريد ارسالها الى حاسوب اخر , فانه اولا يجب ارسالها الى معالجين اثنين ,وهؤلاء يرسلانها الى المعالجين الخاصّين بكل منهما في الخطوة القادمة وهكذا . في ميكانيكية كل مرحلة انه مجرد ان تستلم رسالة ترسل الى معالجين اخرين , وبذلك فان المعالجات تتصل فيما بينها (*communicate*) . ولنبين هنا بعد **n** من مثل تلك المراحل من ارسال الرسائل , ان عدد المعالجات التي ادركت الرسائل هو:

$$1+2+4+\ldots\ldots+2^n \quad \text{or} \quad 2^0+2^1+2^2+\ldots\ldots\ldots+2^n$$

كيف وجدنا المجموع, فلاتوجد هناك خدعة كاستعمال صيغة كاوس التالية:

$$\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$$

مع ذلك اذا جربنا بعدة قيم مختلفة لـ **n** , مثلاً $(n=1,\ 2,\ 3\ldots\ldots)$ نحصل على النمط التالي:

1 + 2 = 3; 1 + 2 + 4 =3 + 4 = 7; 1 + 2 + 4 + 8 = 15 ; 1 + 2 + 4 + 8 + 16 = 31;1 + 2 + 4 + 8 + 16 + 32 = 31 + 32 = 63;

وهكذا . في كل من الحالات السابقة اننا اضفنا المجموع المسبق (*preceding sum*) الى الحد 2^k , حيث نرى ان المجموع المسبق هو(2^k-1). عندما نضيف للمجموع المسبق قيمة 2^k , نحصل على:

$$(2^k-1)+2^k = 2^k+2^k-1 = 2.2^k-1 = 2^{k+1}-1$$

بمعنى اخر ان الصيغة السابقة تصبح:

$$1+2+4+\ldots\ldots+2^n = 2^{n+1}-1$$

من الصيغة اعلاه يمكننا الحصول على سلسلة من الصيغ تعتمد على قيم **k** التي سنعوضها بـ **n** .

وسنبين كيف ان صيغة ما نحصل عليها من الصيغة التي قبلها .بمعنى اخر سنرى كيف الصيغة لـ ($n = k$) تشتق من الصيغة لـ ($n = k - 1$) . اذن على اساس ذلك ان النمط يستمر الى الابد. وبالتالي نستنتج ان الصيغة نافذة لكل قيمة من قيم المتغير **n** . وهذا هو جوهر التفكير وبخاصة الاستنتاج بواسطة الاستنتاج الرياضي.

ان المثال اعلاه يوضح مبدأ الاستنتاج الرياضي (*principle of mathematical induction*) , ولنبين كيف ان الاستنتاج الرياضي يستخدم بطريقة اكثر عمقاً, افرض ان :[1]

S_n ترمز الى مجموع الاعداد الصحيحة الموجبة الاولى (**n**) .

$$S_n = 1 + 2 + 3 + ... + n$$

افرض ان احدا زعم ان العلاقة هي: $S_n = \frac{n(n+1)}{2}$ لقيم $(n = 1, \ 2,.......)$. اذن مجموعة من الجُمل يمكن الحصول عليها كمايلي:

$$S_1 = \frac{1(1+1)}{2} = 1$$

$$S_2 = \frac{2(2+1)}{2} = 3$$

$$S_3 = \frac{3(3+1)}{2} = 6$$

$$\vdots$$

$$\vdots$$

ولو فرضنا ان الجُمل المذكورة اعلاه كانت صحيحة . اذن يجب علينا ان نُبيِّن ان كل الجُمل التي تسبق على سبيل المثال المعادلة نسبة الى $(n+1)st$ هي ايضا صحيحة, ومن ثم لنبرهن بالتالي ان صيغة المعادلة :

$$S_n = \frac{n(n+1)}{2}$$

تحقق المطلوب .فلو كانت الجمل التي تسبق الجملة $(n+1)st$ صـحيحة , فبشـكل خـاص ان المعادلـة ***nth*** صـحيحة ايضاً اي $S_n = \frac{n(n+1)}{2}$ صحيحة. والان يجب ان نبين ان المعادلة $(n+1)st$ هي صحيحة مـن خـلال اسـتبدال قيمة **n** بـ (n+1) :

$$S_{n+1} = \frac{(n+1)(n+1+1)}{2}$$

وطبقاً الى التعريف:

$$S_n = 1+2+3+....+n$$

فان :

$$S_{n+1} = 1+2+3+....+n+(n+1)$$

ونرى ان S_n هي ضمن S_{n+1}, اذن:

$$S_{n+1} = S_n + (n+1)$$

وبربط المعادلتين S_n و S_{n+1} نحصل على :

$$S_{n+1} = S_n + n + 1 = \frac{n(n+1)}{2} + n + 1 = \frac{(n+1)(n+2)}{2}$$

ان في برهاننا هذا استخدمنا الاستنتاج الرياضي بخطوتين : **الاولى** تحققنا من ان الجملـة نسـبة الى (n=1) كانـت صـحيحة , **والثانية** افترضنا ان الجمل (1, 2,.............n) كانت صحيحة وبرهنا اخيراً ان الجملة نسبة الى $(n+1)st$ هي صحيحة ايضاً.حيث ان في برهان الجملة نسبة الى $(n+1)st$ سمحنا باستخدام الجمل (1, 2,.............n) . وان العملية هذه هي اننا استخدمنا الاستنتاج الرياضي لنجعل الجمل (1, 2,.............n) ذات علاقة مع الجملة نسبة الى $(n+1)st$.

ان ماذكرناه اعلاه يعتبر اساساً لوضع مبادئ الاستنتاج الرياضي ***induction*** (***principle of mathematical***) التالية:

افرض ان $S(n)$ تمثل جملة حول المتغير **n** (عدد صحيح موجب), ذلك انها اما صحيحة او خاطئة .

وافرض ان :

1- الجملة $S(1)$ صحيحة True , واذا كانت

2- الجملة $S(i)$ صحيحة لكل القيم $(i < n+1)$, فان الجملة $S(n+1)$ صحيحة ايضاً . وبالتالي فان $S(n)$ صحيحة لكل عدد موجب (n).

Suppose that for each positive integer n we have a statement $S(n)$ *that is either true or false.*
Suppose that:

1- $S(1)$ *is true*

2-if $S(i)$ *is true , for all* $i < n+1$, *then* $S(n+1)$ *is true .*

Then $S(n)$ *is true for every positive integer n.*

ان الجملة (1) ضمن مبادئ الاستنتاج الرياضي الاساسية اعلاه تدعى الخطوة الاساسية (*Basic Step*), والجملة الشرطية تدعى احياناً خطوة الاستنتاج (*Inductive Step*).

او يمكننا الاشارة ايضاً الى مبادئ الاستنتاج الرياضي كما يلي:

افرض أنَّ $S(n)$ تمثل جملة حول المتغير **n** ,

1- فاذا كانت الجملة $S(1)$ صحيحة (**true**).

2- وكانت الجملة $S(k-1)$ تتضمن الجملة $S(k)$ لكل $(k > 1)$, فان الجملة $S(n)$ صحيحة لكل الاعداد الصحيحة الموجبة (n).

Let S(n) be a statement about an integer variable n. If

1-statement S (1) is true, and

2-The statement S(k-1) implies the statement S(k) for each k>1, then the statement S(n) is true for all positive integers n.

تعريف 2-1

يُعرَّف مضروب n (n *factorial*) , $(n!)$, كمايلي:

$$(n!) = \begin{cases} 1 & \text{if} \quad n = 0 \\ n(n-1)(n-2) & \text{if} \quad n \geq 1 \end{cases}$$

وهذا يعني ان اذا كانت $n \geq 1$, فان $(n!)$ تساوي حاصل ضرب كل الاعداد الصحيحة بين **1** و **n** (**n** من ضمنها). كما ان $(0!)$ عُرِّف ليساوي 1 ($(0!) = 1$).

▲

مثال 2-1:

$$0! = 1! = 1, \qquad 3! = 3.2.1 = 6, \qquad 6! = 6.5.4.3.2.1 = 720.$$

مثال 2-2:

استخدم الاستنتاج الرياضي لاثبات ان :

$$n! \;\; \geq 2^{n-1} \qquad \text{for} \quad n = 1,2,.......$$

الحل:

1- الخطوة الاساسية: ***Basic Step*** : يجب اولاً ان نبين ان المعادلة اعلاه صحيحة اذا كانت ($n = 1$), واثبات ذلك سهلاً حيث ان

$$1! = 1 \geq 1 = 2^{1-1} = 2^0 = 1$$

2- خطوة الاستنتاج : ***Inductive Step***

وفيها يجب ان نُبيِّن انه اذا كانت :

$$i! \;\; \geq 2^{i-1} \qquad \text{for} \quad i = 1,.....,n,$$

فان :

$$(n+1)! \geq 2^{n}$$

افــــرض ان:($i! \geq 2^{i-1}$ for $i = 1,.....,n,$) , فانـــــه في حالـــــة خاصــــة $(i = n = 1,2.....)$, يكون لدينا:

$$n! \geq 2^{n-1}$$

وبربط المعادلتين ($n! \geq 2^{n-1}$ و $(n+1)! \geq 2^{n}$) فاننا نحصل على:

$$(n+1)! = (n+1)(n!).$$

والان:

$$\begin{aligned}(n+1)! &= (n+1)(n!).\\ &\geq (n+1)2^{n-1} \quad \text{by} \quad n! \geq 2^{n-1}\\ &\geq 2.2^{n-1} \quad \text{since} \quad n+1 \geq 2\\ &= 2^{n}\end{aligned}$$

والنتيجة تدل على ان المعادلة ($(n+1)! \geq 2^{n}$) هي صحيحة.

وبما ان الخطوة الاساسية وخطوة الاستنتاج قد تحققا , فان مبادئ الاستنتاج الرياضي تخبرنا بصحة المعادلة

$$n! \geq 2^{n-1} \quad \text{for} \quad n = 1,2,.......$$

ولكل عدد صحيح موجب (n) .

مثال 2-3:

بين ان المعادلة : $1 + 2 + 2^{2} + ... + 2^{n} = 2^{n+1} - 1$ صحيحة لكل عدد صحيح موجب (n).

الحل:

نفرض ان $S(n)$ هي الجملة: ($S(n) = 1 + 2 + 2^{2} + ... + 2^{n} = 2^{n+1} - 1$). اذن

$$S(1) = 1 + 2 = 2^{2} - 1$$

تعتبر جملة صحيحة. وهذه الخطوة الاساسية من البرهان بواسطة الاستنتاج الرياضي.. ولغرض ان نبين ان $S(k-1)$ تتضمن $S(k)$, فاننا يجب ان نبين ان عندما $S(k)$ هي صحيحة فان $S(k-1)$ صحيحة ايضاً. وهذه تدعى خطوة الاستنتاج. الان افترض ان $S(k-1)$ هي صحيحة (هذه الجملة تدعى فرضية الاستنتاج (*inductive hypothesis*)).
وعلى اساس ذلك نفترض ان:

$$1 + 2 + 2^2 + \dots\dots\dots + 2^{k-1} = 2^k - 1$$

حيث $(n = k - 1)$.

وهنا يجب ان نُبيِّنَ انَّ $S(k)$ وبالتحديد: ($S(k) = 1 + 2 + \dots\dots\dots + 2^k = 2^{k+1} - 1$) هي صحيحة.

ونلاحظ من هذه المعادلة ان الطرف الايسر منها ينتهي بـ 2^k . اذن نضيف 2^k الى طرفي المعادلة التي حصلنا عليها من $S(k-1)$ لنحصل على مايلي:

$$1 + 2 + 2^2 + \dots\dots\dots + 2^{k-1} + 2^k = 2^k - 1 + 2^k = 2 \cdot 2^k - 1$$
$$= 2^{k+1} - 1$$

ومن ذلك نستنتج ان $S(k)$ صحيحة عندما $S(k-1)$ صحيحة. وبواسطة مبدأ الاستنتاج الرياضي, فان $S(n)$ صحيحة لكل الاعداد الصحيحة الموجبة (n) , اي ان :

$$1 + 2 + 2^2 + \dots\dots\dots + 2^n = 2^{n+1} - 1$$

هي صحيحة لكل عدد صحيح موجب (n) . وهذا الجزء الاخير يدعى نتيجة الاستنتاج (*inclusion inductive*) .
▲

ونلاحظ ان البرهان اشتمل على ثلاثة خطوات . الاولى تمثل الخطوة الاساسية التي تحققت من ان $S(1)$ هي صحيحة. وخطوة الاستنتاج التي تحققت من ان $S(k-1)$ تتضمن $S(k)$. والخطوة الاخيرة تمثل نتيجة الاستنتاج , حيث انه من مبدأ الاستنتاج الرياضي , استنتجنا ان $S(n)$ صحيحة لكل الاعداد الصحيحة الموجبة (n). ونلاحظ ايضاً انه عندما نريد برهان ان :

$$s(n-1) \text{ implies } s(n),$$

فانه يجب ان نبين انه عندما $s(n-1)$ صحيحة فان **s(n)** صحيحة ايضاً. ومن اجل ذلك ان صحة $s(n-1)$ تقودنا الى صحة **s(n)**.

We show that assuming the truth of $s(n-1)$ leads us to conclude the truth of $s(n)$

.

2-2 الاستقراء (الاستنتاج) الرياضي بأساسٍ مختلف.

Math. Induction with a Different Base

لتوضيح ذلك افترض اننا نريد ان نستعمل الاستنتاج الرياضي لبرهنة الجملة التالية:

$$1+2+\dots\dots\dots+n = \frac{n(n-1)}{2}$$

نلاحظ ان الطرف الايسر من المعادلة اعلاة لامعنى له بدون ان تكون n قيمتها تساوي 2 او اكثر . اذن في هذه الحالة نريد ان نبين ان $S(n)$ هي صحيحة لكل الاعداد الصحيحة $(n \geq 2)$ بدلا من ان تكون لكل الاعداد الصحيحة الموجبة (يعني بدلاً من $n \geq 1$). وبما انه نريد برهنة الجمل $S(2)$, $S(3)$, ... اذن سنبدأ من القيمة $(n = 2)$ وهذه تمثل الخطوة الاساسية (*Basic Step*). وللمحاولة في برهنة صيغ اخرى يقودنا بنفس الطريقة الى خطوات اساسية اخرى . ويتلائم ذلك التعدد في الاساس باستخدام نسخة الاستنتاج الرياضي التالية:

افرض ان $S(n)$ تمثل جملة حول المتغير **n** ,

1- فاذا كانت الجملة $S(b)$ صحيحة (true).

2- وكانت الجملة $S(n-1)$ تتضمن الجملة $S(n)$ لكل $(n > b)$, فان الجملة $S(n)$ صحيحة لكل الاعداد الصحيحة $n \geq b$.

Let $S(n)$ be a statement about an integer variable *n*. If

1- $S(b)$ is true, and

2- the statement $S(n-1)$ implies $S(n)$ for all $(n > b)$,

Then $S(n)$ is true for all integer $n \geq b$.

مثال 2-4:

برهن باستخدام الاستقراء (الاستنتاج) الرياضي انه عندما $(n \geq 2)$ فان الجملة التالية صحيحة:

$$1+2+\dots\dots\dots+n = \frac{n(n-1)}{2}$$

الحل:

1- الخطوة الاساسية: نبدأ بـ $(n = 2)$ لان الطرف الايسر من المعادلة اعلاة لامعنى له بدون ان تكون n قيمتها تساوي 2 او اكثر. اذن :

$$1 = 2.1/2 = 1$$

وهو صحيح.

-1 فرضية الاستنتاج: نفترض المعادلة اعلاه صحيحة اذا كانت $(n = k - 1)$, وهذا يعطينا مايلي:

$$1 + 2 + \ldots\ldots\ldots + (k-1) - 1 = \frac{(k-1)(k-1-1)}{2}$$

Or

$$1 + 2 + \ldots\ldots\ldots + k - 2 = \frac{(k-1)(k-2)}{2}$$

وهنا اذا اردنا ان نحصل على الطرف الايسر من الصيغة لـ $(n = k)$, فاننا نحتاج الحد $(k - 1)$ على الطرف الايسر وبذلك يجب اضافة $(k - 1)$ الى طرفي المعادلة لنحصل على مايلي:

$$1 + 2 + \ldots\ldots\ldots + k - 2 + \boldsymbol{k - 1} = \frac{(k-1)(k-2)}{2} + \boldsymbol{k - 1}$$

$$= \frac{k^2 - 3k + 2 + 2k - 2}{2}$$

$$= \frac{k^2 - k}{2} = \frac{k(k-1)}{2}$$

وهذا يُبيِّن تحقيق المعادلة عندما $(n = k)$ وهذا بدوره يكمِّل خطوة الاستنتاج. وبواسطة مبدأ الاستنتاج الرياضي فان الصيغة اعلاه تتحقق لكل قيم $(n \geq 2)$.

ان الصيغة في المثال اعلاه هي حالة خاصة من الصيغة التالية :

$$1 + r + \ldots + r^n = \frac{1 - r^{n+1}}{1 - r}. \quad \text{for} \quad r \neq 1$$

التي يمكن برهنتها لكل الاعداد الصحيحة غير السالبة (*non-negative integers n*) كمايلي:

1- **الخطوة الاساسية:** تمثل الحالة عندما قيمة n تساوي صفراً $(n = 0)$.وبذلك فان برهاننا يشمل كل الاعداد الصحيحة الغير سالبة , اي ليس الاعداد الصحيحة الموجبة فقط. عندما $(n = 0)$ فان الصيغة تنتج:

$$1 = \frac{1 - r}{1 - r}$$

وهي بالتاكيد صحيحة.

2- **فرضية الاستنتاج** : نفترض ان الصيغة صحيحة عندما $(n = k - 1)$. اذن تصبح الصيغة الاصلية كمايلي:

$$1 + r + r^2 + \dots \dots . + r^{k-1} = \frac{1 - r^{k-1+1}}{1 - r} = \frac{1 - r^k}{1 - r}$$

نضيف (r^k) الى طرفي المعادلة (لنحصل على المجموع للصيغة عندما $(n = k)$ وينتج :

$$1 + r + r^2 + \dots\dots\dots + r^{k-1} + r^k = \frac{1 - r^k}{1 - r} + r^k$$

$$= \frac{1 - r^k + r^k - r^{k+1}}{1 - r}$$

$$= \frac{1 - r^{k+1}}{1 - r}$$

والنتيجة تُبين ان المعادلة صحيحة عندما $(n = k)$, وهذا يُكمِّل نتيجة الاستنتاج. وبواسطة مبدأ الاستنتاج الرياضي فان الصيغة اعلاه تتحقق لكل الاعداد الصحيحة الغير سالبة (*all non-negative integers n*)

ان المجموع في الطرف الايسر في هذا المثال يسمى مجموع المتواليات الهندسية (geometric series sum) بنسبة مشتركة r (common ratio *r*) للحدود المتتالية (*consecutive terms*) . وان هذا المجموع يعتبر مهماً في الرياضيات المتقطعة وعلم الحاسوب ان النسبة المشركة يعبر عنها بـ

$$\frac{r^{i+1}}{r^i} = r$$

لكن لو كان المجموع في الطرف الايسر هو كما في المعادلة التالية:

$$a + ar + \dots + ar^n = \frac{a(1 - r^{n+1})}{1 - r}. \quad \text{for} \quad r \neq 1$$

فان النسبة تكون:

$$\frac{a.r^{i+1}}{a.r^i} = r$$

وبرهنة العلاقة الاخيرة هي بنفس طريقة العلاقة في بداية هذا المثال, حيث الخطوة الاساسية تبدا $(n = 0)$ لنحصل على :

$$a = \frac{a(1 - r^{0+1})}{1 - r} = a$$

وهي صحيحة . ونترك خطوة الاستنتاج للقارئ لتكملتها.

مثال 2-5:

استخدم الاستنتاج الرياضي لاثبات ان : $5^n - 1$ ( ) قابلة للقسمة على **4** عندما تكون قيم $(n = 1,2,3....)$

الحل:

1- **الخطوة الاساسية** : عندما $(n = 1)$, فان :

$$5^n - 1 = 5^1 - 1 = 4$$

وهي تقبل القسمة على **4** , اي انها صحيحة.

2- **خطوة الاستنتاج**: نفترض ان $5^n - 1$ قابلة للقسمة على **4**, اذن يجب ان نبين ان $5^{n+1} - 1$ تقبل القسمة على 4 ايضاً, ولبيان وجودعلاقة بين حالة الجملة نسبة الى $(n+1)$ والحالة نسبة الى **n** , فاننا نكتب:

$$5^{n+1} - 1 = 5.5^n - 1 = (1+4).5^n - 1 = (5^n - 1) + 4.5^n$$

وبالفرضية فان $5^n - 1$ قابل للقسمة على **4**, وبما ان 4.5^n هي ايضا تقبل القسمة **على 4** . اذن مجموع الحدّين

$$5^{n+1} - 1 = 5.5^n - 1 = (1+4).5^n - 1 = (5^n - 1) + 4.5^n$$

يقبل القسمة على **4** . **وبتحقق الخطوة الاساسية وخطوة الاستنتاج** , فانه بواسطة مبدأ الاستنتاج الرياضي ان :

$5^n - 1$ قابلة للقسمة على **4** عندما تكون قيم $(n = 1,2,3....)$.

2-3 برهنة المتباينات بواسطة الاستنتاج الرياضي:

Proving Inequalities by Induction

هناك عدد من المتباينات المهمة التي يمكن برهنتها بواسطة الاستنتاج الرياضي . والتطبيق مع تلك المتباينات يساعدنا على فهم الاستنتاج الرياضي. حتى بالرغم من ان خطوة الاستنتاج مازالت تشتمل على عمل شئ(غالباً مايكون اضافة) الى طرفي المعادلة او المتباينة. ومانعمله مع الطرفين هنا ليس تقريباً بنفس الوضوح عما نعمله مع صيغ الجمع. وللوضوح سنبدأ في المثال التالي:

مثال 2-6:

بيِّن ان $(n+1)^2 < 2n^2$ صحيحة لكل الاعداد الصحيحة اكبر او تساوي 3.

Show that $(n + 1)^2 < 2n^2$ for all integers $n \geq 3$.

الحل:

الخطوة الاساسية : نبدأ باختيار $(n = 3)$. فاذا كانت $(n = 3)$, فانالصيغة تقرأ

$$(3+1)^2 < 2.3^2, \text{or } 16 < 18,$$

وهي صحيحة .

خطوة الاستنتاج: ونبدأها بفرضية الاستنتاج وهي ان نفرض المُتباينة صحيحة عندما ($n = k - 1$). لكي نحصل على:

$$(k - 1 + 1)^2 < 2(k - 1)^2 \quad or \quad k^2 < 2(k^2 - 2k + 1) = 2k^2 - 4k + 2$$

والان نقارن المُتباينة للحالة $(n = k - 1)$ والحالة $(n = k)$ لاننا نريد ان يكون الطرف الايسر $(k + 1)^2$ لكي تكون لدينا علاقة مع $n = k$ بدلاً من $(n = k - 1)$. وبما ان :

$$(k + 1)^2 = k^2 + 2k + 1$$

اذن يبدو انه يجب ان نضيف $2k + 1$ الى طرفي المُتباينة:

$$k^2 < 2k^2 - 4k + 2.)$$

لنحصل على :

$$k^2 + 2k + 1 < 2k^2 - 4k + 2 + 2k + 1 = 2k^2 - 2k + 3$$

وبما اننا نعمل فقط مع قيم k التي هي اكبر من 4 بسبب ان $(n = k - 1)$ وان $n \geq 3.$ تعطينا $k \geq 4.$ اذن . $2k$ اكبر من 3 . لذا $(-2k + 3)$ هو سالب . اذن :

$$k^2 + 2k + 1 < 2k^2 - 2k + 3 < 2k^2 \quad \text{or} \quad (k + 1)^2 < 2k^2$$

وبالتالي عندما المُتباينة صحيحة لـ $(n = k - 1)$ و n اكبر من 3 ($\text{and } n \geq 3$) , فان المُتباينة صحيحة لـ $(n = k)$. اذن المُتباينة صحيحة لكل الاعداد الصحيحة ($n \geq 3$).

مثال 2-7 متروك حله للقارئ:

بين ان ($(n + 1)^2 < 2n^2$) صحيحة لكل قيم ($n \geq 3$) , مستخدماً النسخة الجديدة من مبدأ الاستنتاج الرياضي الانفة الذكر.

Show that $(n + 1)^2 < 2n^2$ for $n \geq 3$, using thus new version of the principle.

4-2 تطبيق الاستنتاج الرياضي في علم الحاسوب

Application of Induction in Computer Since

1-4-2 الخوارزميات التكرارية في الحساب *Recursive Algorithms in Computing*

ان دالة المضروب (*factorial function*) تُعرَّف احياناً بالقول ان $(n!)$ هي حاصل ضرب الاعداد الصحيحة الموجبة الاولى (*first n positive integers*) . يمثل هذا التعريف على سبيل المثال نقول $(3!= 3.2.1)$. بينما التعريف الاستنتاجي او الاستقرائي (*inductive definition*) لدالة المضروب factorial هو:

تعريف 2-2

يُعرف مضروب n (*n factorial*) , $(n!)$, كمايلي:

$$(n!)=\begin{cases}1 & \text{if} \quad n=0\\ n(n-1)(n-2) & \text{if} \quad n\geq 1\end{cases}$$

وهذا يعني ان اذا كانت $n \geq 1$, فان $(n!)$ تساوي حاصل ضرب كل الاعداد الصحيحة بين **1** و **n** (**n** من ضمنها). كما ان $(0!)$ عُرِّف ليساوي 1 ($(0!)=1$).

مثال 2-8:

احسب مضروب **0 , 3 ,6** من خلال استخدام التعريف الاستنتاجي لدالة المضروب:

$$0!= 1!= 1, \qquad 3!= 3.2.1 = 6, \qquad 6!= 6.5.4.3.2.1 = 720.$$

ان التعريف الاستنتاجي لدالة المضروب يعطينا طريقة لحساب $(n!)$ التي يمكن برمجتها مباشرة في لغات البرمجة الحديثة . كما سنرى في البرامج التالية : حيث البرنامج الاول يرينا مقطع لبرنامج الحاسوب مكتوب بلغة البيسك (**Basic**) والثاني بلغة باسكال (Pascal) والثالث بلغة السي ++ (C++) التي كلها تحسب دالة المضروب المسماة (**fact**) والمُعطاة كمايلي:

fact (0) = 1

fact (n) = $n \cdot fact$ (n – 1)   if n > 0

1-Basic Factorial Function

```
DEF fact (n)
IF n = 0 THEN LET fact = 1
ELSE LET fact = n*fact (n - 1)
END IF
END DEF
```

2-Pascal Factorial Function

```
function Fact (n : integer):integer;
begin
if (n = 0)
then Fact : = 1
else Fact : = n*Fact (n – 1)
End;
```

3-Factorial function using C++ language:

```
#include<iostream.h>
        int fact(int);
        int main()
        {
        cout<<"enter a number\n";
        int x;
        cin>>x;
        cout<<fact(x);
        return 0;}
```

```
int fact(int x){
if (x==0 || x==1)
return 1;
else
return x*fact(x-1);}
```

- ***Example*: *factorial***

  $n! = n * (n – 1) * (n – 2) * … * 1$

  – ***Recursive relationship*** ($n! = n * (n – 1)!$)

    $5! = 5 * 4!$

    $4! = 4 * 3!…$

  – **Base case** ($1! = 0! = 1$)

No recursively definition: اما التعريف غير التكراري للمضروب فهو

```
fact=1;

for ( int counter = number; counter >= 1; counter-- )
  fact *= counter;
```

ان الامثلة اعلاه هو مانسميه البرمجة التكرارية الذاتية (*recursive programming*) والتي تعني كتابة برنامج الحاسوب باستخدام تعريف الاستنتاج. فلو سألنا الحاسوب لاستعمال الدوال اعلاه في الامثلة المذكورة لحساب مضروب 3 (3!) , فان الحاسوب سيعمل بالضبط نفس الحسابات التي وردت في تلك الامثلة. كما ان اسلوب التكرارية يعطينا طريقاً سهلاً للتعبير عن اية خوارزمية التي تقلل المسالة الى شكل مبسط لنفس المسألة الاصلية . بعض تلك المسائل لها خوارزميات غير تكرارية (*non-recursive*) والتي تعتبر سهلة في الفهم وتحتاج عمل اقل .

ان المثال القادم يُبيِّن ان التكرار بالرجوع (*recursion*) غير محدد لحل المسائل العددية. فمثلاً ان خوارزمية التكرار بالرجوع (*recursive algorithm*) لايجاد اصغر عنصر في قائمة ما, تبنى على الفكرة التالية:

ان اصغر عنصر في قائمة, اما ان يكون هو العنصر الاول فيها او العنصر بين الموقع الثاني ونهاية القائمة.

مثال 2-9:

اكتب خوارزمية التكرارالذاتي (*recursive algorithm*) لايجاد اصغر عنصر في القائمة L من الاعداد بين الموقع i والموقع j (داخل ضمنها). استخدم $L(k)$ لتقوم مقام المدخل في الموقع k من القائمة.

الحل:

اسم الخوارزمية :خوارزمية ايجاد الاصغر

المدخلات :*inputs* : قائمة **L** من الاعداد والموقعين **i** و **j** في تلك القائمة.

المخرجات *outputs* : الموقع **k** للرقم الاصغر في القائمة بين الموقعين **i** , **j** .

الاجراء : *procedure* :

1- اذا كانت $(i = j)$, فاننا نفرض $(k = i)$.

2- اذا كانت $(i \neq j)$, نستعمل خوارزمية ايجاد الاصغر الى المواقع $(i+1)$ وصولاً الى **j**.وهذا يعطي **k** .

3- نبدأ المقارنة هنا: اذا كانت $L(i)$ اصغر من $L(k)$ "$(L(i) < L(k))$" , فاننا نغير **k** الى **i** . لكن اذا كانت اذا كانت $L(i)$ ليست , فلانعمل الخوارزمية شيء.

مثال 2-10:

صف الخطوات التي تُؤخذ في خوارزمية ايجاد الاصغر الى القائمة 3, 1, 2

1- نبدأ بـ $(i = 1)$ و $(j = 3)$. بما ان $(i \neq j)$, نستخدم خوارزمية الاصغر مع $(i = 2)$ و $(j = 3)$.

2- مع $(i = 2)$ و $(j = 3)$. بما ان $(i \neq j)$, نستخدم خوارزمية الاصغر مع $(i = 3)$ و $(j = 3)$.

3- مع $(i = 3)$ و $(j = 3)$. , افرض $(k = 3)$. وبما اننا نستخدم خوارزمية التكرار الذاتي (*recursive algorithm*) لحساب الاصغر . اذن نرجع من الخطوة الاخيرة الى التي قبلها, اي الحالة $(i = 2)$.

4- نعود لنكمل الحالة $(i = 2)$, نقارن $L(2)$ مع $L(3)$ ونغيّر k الى 2. وبما ان واحد اصغر من 2 ($1 < 2$) , هذا يعني ان $L(2)$ اصغر من $L(3)$,"$L(2) < L(3)$". اي ان $L(2) = 1$ و $L(3) = 2$.

5- نرجع لنكمل الحالة $(i = 1)$, نقارن $L(1)$ مع $L(2)$ اي 1 مع 3 (انظر القائمة في المثال). نترك $(k = 2)$ و بعد ذلك فان $L(1) > L(2)$. وقد تبين لدينا ان الرقم الاول اكبر من الثاني في القائمة والثاني اصغر من الثالث فيها .اذن الرقم الثاني هو الاصغر .

تمارين الفصل الثاني

1- Using principle of mathematical induction to prove that :

$1+2+2^2+....+ 2^n = 2^{n+1}-1$ for each positive integer n.

2- Use the induction to show that $n! \geq 2^{n-1}$ **for n=1,2,…………**

3- Use mathematical induction to show that if $r \neq 1$,

$$a + ar + ar^2 + ... + ar^n = \frac{a(r^{n+1} - 1)}{r - 1}$$ for all non-negative integers *n*.

4- Write down an induction definition of the factorial function ($n!$).Then compute 5! from the induction definition.

5- If y is the sequence defined by $y_n = 2^{-n}$, $-1 \leq n \leq 3$, **find the elements of y.**

الفصل الثالث

المجموعات والعلاقات والدوال ورموزها

Sets, Relations and functions and Notations

3-1 مقدمة:

المجموعة **Set** : هي مجموعة من الاشياء (a Set of objects) .وعلاقة الرياضيات المتقطعة بالمجموعات هي ان الرياضيات المتقطعة تتعلق بتراكيب مثل رسم المخططات (*Graphs*) والتي تعني مجموعة (*Set*) من (النقاط *vertices* والحدود *edges*) وتتعلق ايضا بالجبر البولياني - *Boolean algebra* (والذي هو عبارة عن مجموعات مع عمليات يتم تعريفها عليها) كما سنتطرق الى ذلك لاحقاً. والمجموعات تنتمي الى مايسمى لغة الرياضيات *The language of* (*mathematics* .والمجموعة لاتأخذ الترتيب لعناصرها بنظر الاعتبار.

وبخلاف المجموعة فان المتتالية (*a sequence*) تأخذ الترتيب بنظر الاعتبار . فقائمة من الحروف كالتي تظهر في الكلمة هي مثال على المتتالية . ويتضح من ذلك ان الترتيب مهم جدا, فعلى سبيل المثال (*form*) و (*from*) كلمات مختلفة .

اما العلاقة الرياضية (*relation*) فهي مجموعة من الازواج المرتبة (*ordered pairs*). ان وجود الزوج المرتب (**a, b**) في علاقة رياضية يفسر كمؤشر على وجود علاقة (*relationship*) من **a** الى **b** . ان نموذج قواعد البيانات العلائقية (*Relational Database Model*) , التي تساعد المستخدمين في الدخول الى البيانات في قاعدة البيانات (التي هي مجموعة من السجلات (*records*) المعالجة بواسطة الحاسوب) اسست على اساس مبدأ العلاقة (*relation*).

من جانب اخر, ان الدالة (*function*) التي هي حالة خاصة من العلاقة الرياضية(*relation*) تعطي لكل عنصر من المجموعة **X** عنصر واحد بالضبط من المجموعة **Y** . ان الدوال تستخدم بشكل واسع في الرياضيات المتقطعة , على سبيل المثال , الدوال تستخدم لتحليل الوقت الضروري لتنفيذ الخوارزميات *Algorithms* .

ومن ذلك نرى ان المجموعة set هي الاساس لكل من الرياضيات والتطبيقات الرياضية . وهي ببساطة اي مجموعة من الاشياء كما ذكرنا .,فاذا كانت المجموعة محددة وغير كبيرة , فاننا نستطيع وصفها بترتيب عناصرها من خلال ادراجها بقائمة . فمثلاً المعادلة التالية :

$$A = \{1,2,3,4\}$$

تصف مجموعة A على انها مكونة من اربعة عناصر هي 1, 2, 3, و 4 . ان المجموعة A حددت بعناصرها وليس بأي ترتيب خاص لعناصرها , وبذلك نستطيع كتابة المجموعة A بشكل اخر(ترتيب اخر) كما في:

$$A = \{1,3,4,2\}$$

اما اذا كانت المجموعة كبيرة ومحددة او غير محددة , فاننا نستطيع وصفها بادراج خاصية على شكل قائمة *(listing a property)* ضرورية لاعضاء المجموعة *(membership)* . فعلى سبيل المعادلة :

$$B = \{x \mid x \text{ is apositive, even integer}\}$$

تصف المجموعة **B** مكونة من كل الاعداد الصحيحة الزوجية الموجبة , بمعنى ان **B** تشتمل على الاعداد الصحيحة **2, 4, 6**, وهلم جرا. ويمكن قراءتها بـ(ان **B** تساوي المجموعة لكل القيم x , حيث x هي الاعداد الصحيحة الموجبة). ان الخاصية *(property)* هنا مهمة لوصف المجموعة وهذه الخاصية هي: (موجبة واعداد زوجية صحيحة : *(is a positive ,even integer)* . لاحظ ان الخاصية تظهر بعد الرمز (|) .

من جانب اخر اذا كانت X مجموعة محددة *(if x is a finite set)* , فان عدد عناصرها نفرضه بـ $|X|$:

$$|X| = \text{number of elements in } X$$

فلو اعطينا وصفاً لـ **X** كما وصفنا المجموعة **A** والمجموعة **B** اعلاه وكذلك اعطينا قيمة للعنصر x , فاننا نستطيع ان نحدد فيما لو كان العنصر **x** ينتمي او لاينتمي لـ **X** . فلو كانت $x = 1$ فانها تنتمي لـ A و لاتنتمي لـ B وهكذا..

يقال لمجموعتين مثل X و Y انهما متساويتين ونكتبهما X=Y اذا كانت X و Y لهما نفس العناصر , ويمكن التعبير عنهما بصيغة اخرى X=Y اذا عندما x تنتمي لـ X ($x \in X$) ,و انها بنفس الوقت ايضا تنتمي لـ Y ($x \in Y$) , والعكس صحيح ايضاً.

تعريف 3-1:

المجموعة (Set) :هي مجموعة من الاشياء معرفة بشكل واضح تسمى عناصر او اعضاء تلك المجموعة :

A set is a well-defined collection of objects called elements, or members, of the set.

ان الخاصية الرئيسة لمجموعة (*Set*) في الرياضيات هي انها تكون واضحة. هذا يعني اعطاء اي شئ انه يجب ان يكون واضحاً فيما اذا ذلك الشئ هو عنصر من المجموعة. وبالتالي اذا اعتبرنا مجموعة الاعداد الكاملة الزوجية (*even whole numbers*) , فاننا نعرف ان اي عدد زوجي مثل **0**, **2**, **4**, ... هو عنصر لتلك المجموعة لكن ليس الا. وبذا فان مجموعة الاعداد الزوجية الكاملة(*even whole numbers*) واضحة.

اننا نستخدم الحروف الانكليزية الكبيرة مثل ($A,B,C,X,Y \quad and \quad Z$) للدلالة على المجموعات , ونستخدم الحروف الصغيرة مثل ($a,b,c,x,y \quad and \quad z$) للدلالة على عناصر المجموعة . كما انه من المألوف ان نضع عناصر المجموعة محصورة بين قوسين{ } ومفصولة فمابينها بفواصل (,) *commas* . هكذا فان $A = \{1,2,3,4\}$ تمثل مجموعة اسمها A وعناصرها $4,3,2,1$ () . وللدلالة على ان **2 او 4** هي عنصر من عناصر A او انها في المجموعة **A** , فاننا نكتب :

$$4 \in A$$

وتقرأ **4** هي عنصر من عناصر المجموعة A :" *4 is an element of the set A*" .

ولبيان ان 6 ليست من عناصر المجموعة A اي لاتنتمي لها , فاننا نكتب :

$$\neg(6 \in A) \text{ او } 6 \notin A$$

2-3 وصف المجموعة :Describing Set

المجموعات يمكن تعريفها بثلاثة طرق:

1- بواسطة اعطاء وصف لفظي (نص) للمجموعة : *By giving a verbal description of the set*

2- بواسطة ادراج عناصر المجموعة بقائمة او جدول :*By listing the element of the set*

3- استخدام رموز مايعرف بـ *Using Set-Builder Notation*

لاحظ الامثلة التالية

Description	list	set-builder
1-The set of counting numbers less than 5 is denoted by	{1, 2, 3, 4}	$\{x \mid x \in I \wedge 1 \leq x < 5\}$ مجموعة اعداد الحساب اقل من 5
2-The set of natural earth satellites	{Moon},	مجموعة اقمار الارض الطبيعية
3-The set of even counting numbers	{2, 4, 6, . . .}	وهنا تعتبر مجموعة *it is infinite set.* غير محددة والنقاط الثلاث تعني القائمة مستمرة ليس لها نهاية . The three dots mean the list goes on without end
4-The set of odd counting number less than 15	{1, 3, 5, . . . 13}	*it is finite set* وهذه تعتبر مجموعة محددة . تمثل مجموعة اعداد العد الفردية اقل من 15 : *The three dots mean the odd numbers after 5 and before 13 are in the set but not listed*

في حالة استخدام الرموز(*set–builder notation*) لتعريف المجموعات نستخدم خاصية دقيقة لوصف المجموعة , فالرمز(|) يعني مثل هذا (***such that***) , لذا فالمجموعات السابقة يمكن كتابتها كمايلي:

وهذه تعتبر طريقة واضحة اخرى لوصف المجموعة والتي تعطي جملاً لوصف عناصرها , وفيمايلي وصف المجموعات اعـلاه باللغة الانكليزية :

{x | x is a counting numbers less than 5} $\{x \mid x \in I \wedge 1 \le x < 5\}$	**Read**, "*the set of all elements x, such that x is a counting number less than 5.*"
{x | x is natural Earth satellite }	**Read**, "*the set of all elements x, such that x is a natural Earth satellite.*"
{x | x is an even a counting numbers }	Read, "*the set of all elements x, such that x is an even counting number.*"
{x | x is an odd a counting numbers less than 15 }	**Read**, "*the set of all elements x, such that x is an odd counting numberless than 15.*"

مثال 3-1:

-1 مجموعة الاعداد الصحيحة اكبر من 10 . ونحددها كمايلي:

$$\{x \mid x \in I \wedge x > 10\}$$

-2 مجموعة الاعداد الزوجية الصحيحة.

$$\{x \mid \exists y[y \in I \wedge x = 2y\}$$

-3 مجموعة الاعداد الصحيحة مضاعفات العدد 3 .

ونحددها بـ $\{3x \mid x \in I\}$ ولايمكن تحديدها بـ $\{x \mid \exists y[y \in I \wedge x = 3y\}$

-4 مجموعة الاعداد النسبية (*rational numbers*) .

$$\{x / y \mid x, y \in I \wedge y \neq 0\}$$

-5 مجموعة كل الاعداد الزوجية الصحيحة غير السالبة واقل من 10:

$$\{0,2,4,6,8\}$$

و الامثلة من (1-5) اعلاه نوضحها باللغة الانكليزية كمايلي:

Examples:	*Sets*
1-*The set of integers greater than 10 is specified as*	$\{x \mid x \in I \wedge x > 10\}$
2-*The set of even integers*	$\{x \mid \exists y[y \in I \wedge x = 2y\}$
3-*The set of integers multiple of 3 is specified by*	$\{3x \mid x \in I\}$ *rather than* $\{x \mid \exists y[y \in I \wedge x = 3y\}$
4-*The set of rational numbers is specified by*	$\{x / y \mid x, y \in I \wedge y \neq 0\}$
5-*The set of which contains all the even numbers, nonnegative integers less than 10*	{0, 2, 4, 6, 8}

مثال 3-2:

اكتب وصفاً لفظياً (*verbal description*) للمجموعات التالية:

a. **{a, b, c, . . . , z }** **b.** **{1, 3, 5, . . . }** **c.** **{3, 6, 9, . . . ,27 }**

أ) مجموعة الحروف بالابجدية الانكليزية.

a. The set of letters in the English alphabet.

ب) مجموعة اعداد الحساب الفردية.

b. The set of odd counting numbers.

ج) مجموعة اعداد الحساب التي هي من مضاعفات الـ 3 واقل من 28 (او 29 او 30).

c. The set of counting numbers that are multiples of 3 and less than 28(or 29, or 30).

مثال 3-3:

اكتب المجموعات التالية مستخدماً طريقة القوائم *listing method* وطريقة *set-builder notations* :

1- مجموعة الارقام في الرقم 1896.

a) The set of digits in the number 1896

2- مجموعة ارقام الحساب الفردية واقل من 6.

b) The set of odd counting numbers greater than 6

3- مجموعة اعداد الحساب اكبر من صفر واقل من 1

c) The set of counting number greater than 0 and less than 1.

4- مجموعة اعداد الحساب مضاعفات العدد اربعة.

d) The set of counting number that are multiples of 4

الحل:

List	*Set-builder notation*
a. {1, 8, 9, 6}	{x| x is a digit in the number 1896 }
b. {7, 9, 11 . . .}	{x| x is an odd counting number and $x > 6$ } The symbol $>$ means “grater than.”
c. { }	{x| x is counting number and $0 < x < 1$} The symbol” $<$ ” means “less than.”
d. { 4, 8, 12, . . . }	{x| x is a counting number that is a multiple of 4}

ان المجموعة بدون عناصر كما هو في الجزء (c) اعلاه يرمز اليها بالرمز $\{\ \}$ or $\varnothing$.

تعريف 2-3 :

ان الرمز $\{\ \}$ or $\varnothing$ يمثل المجموعة الفارغة *(empty)* او الصفرية *(null)* او الخالية (*void*) :

The symbol { } or Ø represents the empty, null, or void set

لاحظ ان المجموعة $\{\phi\}$ ليست فارغة بسبب انها تحوي العنصر ϕ.

3-3 تساوي المجموعات Equality of Sets

بوضوح, ان الترتيب الذي فيه عناصر المجموعة تكون بشكل قائمة لايؤثر في العضوية في المجموعة , وبالتالي اذا سؤلنا في ان نكتب مجموعة الارقام في السنة التي غزت فيها امريكا العراق , فاننا نكتب $\{2,0,0,3\}$, ربما شخص اخر يكتبها $\{3,0,0,2\}$, فكلاهما صحيح . اذن المجموعتان متساويتان .

$$\{2,0,0,3\} = \{3,0,0,2\}$$

وبشكل متشابه فان :

$$\{a,b,c,d,e\} = \{e,d,c,b,a\}$$

$$\{1,3,2\} = \{1,2,3\} \text{ and } \{20, ½\} = \{½, 20\}.$$

وبشكل عام فان مجموعتان تكونا متساوتين , يرمز لهما بـ A=B , اذا كان لهما نفس العناصر (حيث ليس ضرورياً ان يدرجوا بنفس الترتيب).

Definition: *in general form, two sets are equal, denoted by A = B ,if they have the same members (not necessarily listed in the same order).*

ومما يجلب الانتباه ايضاً ان تكرار ادراج العنصر في المجموعة لايؤثر في العضوية للمجموعة. فمثلاً مجموعة الارقام في السنة التي احتل فيها العراق هي $\{2,0,0,3\}$, ويمكن كتابتها ايضاً مثل $\{2,0,3\}$, وبذلك فان $\{2,0,0,3\} = \{2,0,3\}$. وبشكل مشابه فان :

$$\{a, a, b, b, c, c\} = \{a, b, c\}.$$

حيث بالعرف اننا لانضع العنصر المكرر في المجموعة اكثر من مرة واحدة فيها.

مثال 3-4 :

المجموعة $\{x \mid x = 0\}$ والمجوعة $\{x \mid x \in I \wedge -1 < x < 1\}$ متساويتان:

$$\{x \mid x = 0\} = \{x \mid x \in I \wedge -1 < x < 1\}$$

مثال 3-5:

اذا كانت : $A = \{x \mid x^2 + x - 6 = 0\}$ و $B = \{2,-3\}$ فان A=B وذلك لان التعويض بعناصر **B** في **A** يحقق المعادلة في **A** .

3-4 المجموعات الجزئية Subsets

يحدث احياناً ان كل عناصر المجموعة **A** هي ايضاً عناصر مجموعة اخرى **B** , فعلى سبيل المثال اذا كانت **A** تمثل مجموعة الطلاب في صفك المدرسي, و **B** تمثل مجموعة كل الطلاب في المدرسة . ففي هذه الحالة ان كل عنصر في **A** هو ايضاً في **B** , لان كل طالب في صفك هو طالب في مدرستك . في مثل هذه الحالة نقول ان المجموعة **A** هي جزء من المجموعة **B** . ونرمز لذلك بدلالة :

$$A \subseteq B$$

وتقرأ A مجموعة جزئية لـ **B** : *A is a subset of B*

من جانب اخر افرض ان **X** و **Y** مجموعتان , فاذا كان كل عنصر في **X** هو عنصر من **عناصر Y** , فاننا يمكننا القول ان **X** هي مجموعة جزئية من **Y** , ونعبر عن ذلك بـ $X \subseteq Y$.

تعريف 3-3:

المجموعة **A** جزء من **B** ويرمز لها بـ$A \subseteq B$, اذا كان كل عنصر في **A** هو ايضاً عنصر في **B**

The set of A is a subset of B (denoted by $A \subseteq B$ *) if every element of A is also an element of B.*

مثال 3-6:

اذا كانت $A = \{a,b\}$, $B = \{a,b,c\}$ و $C = \{b\}$, فان ذلك يعني ان $A \subseteq B$, $C \subseteq A$ و $A \subseteq A$

وكنتيجة لتعريف المجموعة الجزئية ان $A = B$ عندما $A \subseteq B$ و $B \subseteq A$, واكثر من ذلك ان لأية مجموعة **A** , بما انه كل عنصر لـ **A** هو عنصر **A** , فاننا نحصل على

$$A \subseteq A$$

اذن تعريف المجموعة الجزئية يمكن صياغته كما يلي:

المجموعة **A** هي مجموعة جزئية من **B** اذا هناك عنصراً ليس لـ **A** و انه ايضا ليس عنصراً لـ **B**

the set of A is a subset of B , if there is no element of A that is not an element of B

ومن هذا نستطيع ان نشتق ان $\phi \subseteq A$ بسبب انه العنصرالغير موجود في ϕ انه ليس في A ايضاً.هذا يعني ان المجموعة الخالية (*Empty set*) هي مجموعة جزئية من اية مجموعة .

من جانب اخر ان مجموعة **A** يقال عنها مجموعة جزئية تامة (*a proper subset*) لـ **B** , ويعبر عنها بـ

$A \subset B$, اذا كانت **A** مجموعة جزئية من **B** وهناك على الاقل عنصر واحد لـ **B** غير موجود في **A** . بتعبير اخر , ان $A \subset B$ تعني ان كل عناصر **A** موجودة في **B** لكن **B** تحتوي على الاقل عنصر واحد غير موجود في A . على سبيل المثال: اذا كانت $B = \{1,2\}$, فان المجموعات الجزئية التامة (*a proper subset*) لـ **B** هي:

$$\phi, \{1\}, \text{and} \quad \{2\}$$

لكن المجموعة $\{1,2\}$ نفسها هي ليست مجموعة جزئية تامة لـ B لأن جميع عناصرها هي في B .

5-3 ايجاد المجموعات الجزئية: Finding Subsets

في كل يوم من المناقشات نحن اعتيادياً مدركون للجموعة الكونية لما نتحدث عنه وهو مانعني به مجموعة كل الاشياء التي نتحدث عنها . وبالتعامل مع المجموعات , فان الكون الذي نتحدث عنه يدعى المجموعة الكونية (*universal set*) .

تعريف 3-4:

ان المجموعة الكونية U (Universal Set) هي مجموعة كل العناصر تحت المناقشة او البحث او الدراسة.

The universal set U is the set of all elements under discussion

وبالتالي اذا اتفقنا ان نبحث او نناقش الحروف باللغة الانكليزية , فان المجموعة الكونية U هي:

$$U = \{a, b, \ldots\ldots, z\}$$

ومن ناحية اخرى اذا اردنا ان نبحث او نناقش اعداد الحساب (*counting numbers*), فان مجموعتنا الكونية هي:

$$U = \{1,2,3\ldots\ldots\}$$

مثال 3-7:

اوجد كل المجموعات الجزئية (*All subsets*) للمجموعة $U = \{a, b, c\}$.

الحل:

يجب علينا ان نصيغ المجموعات الجزئية للمجموعة **U** وذلك بتعيين بعض , او عدم تعيين , او كل العناصر لـ **U** الى تلك الجموعات الجزئية وكمايلي:

أ- صيغة كل المجموعات الجزئية بدون عناصر هي ϕ .

ب- صيغة كل المجموعات الجزئية بعنصر واحد هي $\{a\}, \{b\}, \{c\}$

ت- صيغة كل المجموعات الجزئية بعنصرين هي $\{a, b\}, \{a, c\}, \{b, c\}$

ث- صيغة كل المجموعات الجزئية بثلاثة عناصر هي $\{a, b, c\}$.

لاحظ ان المجموعة U في هذا المثال فيها ثلاثة عناصر , وبذا ستكون مجموعاتها الجزئية 8 مجموعات اي ($2^3 = 8$) , كما ان مجموعة مثل $\{a, b\}$ سيكون لها **4** مجموعات جزئية وبالاسم هي: $\phi, \{a\}, \{b\}, \{a, b\}$.

وبمعنى اخر ان مجموعة لها *n* من العناصر يكون عدد مجموعاتها الجزئية هو 2^n

مثال 3-8:

اذا كانت $A = \{a_1, a_2, \ldots\ldots\ldots a_8\}$, فكم مجموعة جزئية فيها ؟

الحل:

بما ان المجموعة **A** فيها ثمانية عناصر , اذن مجموعاتها الجزئية هي 256 مجموعة جزئية ($2^8 = 256$).

مثال 3-9:

حدد كل المجموعات الجزئية التامة (*proper subsets*) للمجموعة $X = \{1,2,3\}$ ومجموعة الاس (set power) .

الحل:

المجموعات الجزئية التامة هي: $\phi, \{1\}, \{2\}, \{3\}, \{1,2\}, \{1,3\}, \{2,3\}$, لاحظ ان المجموعة التي لها n من العناصر , يكون لها $2^n - 1$ من المجموعات الجزئية التامة (*proper subsets*). وعدد عناصر X هنا تحسب من:

$$|X| = n = 3$$

ان المجموعة لكل المجموعات الجزئية (تامة او غير تامة) للمجموعة **X** , يرمز لها بـ $P(X)$ تدعى بـ مجموعة الاس (***power set***) لـ **X** . وعدد عناصرها يحسب من المعادلة التالية:

$$|P(X)| = 2^n = 2^3 = 8$$

اذن عناصر $P(X)$ هي نفسها المجموعة الجزئية التامة مضافاً لها المجموعة $\{1,2,3\}$, اي:

$$\phi, \{1\}, \{2\}, \{3\}, \{1,2\}, \{1,3\}, \{2,3\}, \{1,2,3\}$$

3-6 المجموعة الحقيقية والمتكافئة Truth set and equivalence

اذا كانت $U = \{1,2,3,4,5,6\}$ هي مجموعة الاعداد **من 1 الى 6** , وكانت p تمثل الجملة:

" العدد هو زوجي " : *"the number is even"* , فانه طبقاً لتلك الجملة ان المجموعة الحقيقية (*truth set*) هي:

$$P = \{2,4,6\}$$

اذن **P** هي مجموعة جزئية (*subset*) من المجموعة **U** التي لها الجملة p صحيحة . لاحظ ان **P** (حرف كبير) تعني مجموعة وان p (حرف صغير) تعني جملة .

-1 اننا نستعمل جملة المجموعة *Set Statement* التالية:

$$(x > 1) \wedge (x < 5)$$

حول الاعداد الصحيحة في برنامج الحاسوب , ولو استخدمنا الجملة التالية :

$$p(x) = (x > 1) \wedge (x < 5)$$

فانها تعني مجموعة الارقام $\{2,3,4\}$ وان الجملة حول المتغير x . ويمكن لأخر ان يكتبها بمايلي:

$$(x = 2) \vee (x = 3) \vee (x = 4)$$

اي يمكن ان نكتبها بـ

$$q(x) = (x = 2) \vee (x = 3) \vee (x = 4)$$

ان تلك الجملتين المركبتين $p(x)$ و $q(x)$ صحيحتان ,و بالضبط لهما نفس الحال , اي كل قيمة لـ x تجعل p(x) صحيحة فانها ايضاً تجعل q(x) صحيحة , والعكس صحيح ايضاً. وبالتالي فانهما متكافئتان

$$p\ and\ q\ are\ equivalent: \; p(x) \Leftrightarrow q(x)$$

من جانب اخر فان كل القيم لـ x في كوننا التي تجعل p(x) صحيحة , تسمى المجموعة الحقيقيـة truth set لــ p . ومن ذلك فان :

جملتين حول المتغيرات متكافئتان اذا كان لكل منهما بالضبط نفس المجموعة الحقيقية (على سبيل المثال نفس العناصر) .

فمثلا لو كانت p تمثل الجملة :

" النتيجة تشتمل على وجه عملة معدنية واحد " : *the result has one head*

حول الكون لنتائج نقرتين للعملة , فان المجموعة الحقيقية (*truth set*) (P) للجملة ***p*** هي :

$$\{HT, TH\}.$$

و لو كانت ***q*** تمثل الجملة :

" النتيجة تشتمل على وجهي عملة معدنية " : *the result has two heads*

فان المجموعة الحقيقية (*truth set*) (Q) للجملة ***q*** هي :

$$\{HH\}$$

كما ان تقليبين لعملة معدنية واحدة: (*two coin flips*) نحصل على النتيجة ($probability = 2^2 = 4$)

$$\{HH, HT, TH, TT\}$$

وتقليب العملة ثلاث مرات نحصل على ثمانية احتمالات :($probability = 2^3 = 8$) كما يلي:

1- المجموعة الحقيقية لوجهي العملة المكتوبة (*Truth set P for two heads*) هي:

HHH HHT HTH THH HTT THT TTH TTT

2- المجموعة الحقيقية لوجهي العملة الغير مكتوبة : *Truth set Q for two tails*

HHH HHT HTH THH HTT THT TTH TTT

ومن خلال المجموعتان الحقيقيتان اعلاه نرى ان :

$$P = Q$$

7-3 عمليات المجموعة Set Operations

1-7-3 تقاطع واتحاد المجموعات :Intersections And Unions

تعريف 3-5 : اذا كانت **A** و **B** مجموعتان , فان تقاطع **A** و **B** يعبر عنه بدلالة :

$$A \cap B$$

ويقرأ **A** تقاطع **B** (A intersections B) , ويعني انه المجموعة لكل العناصر المشتركة الى كلا المجموعتين **A** و **B** .
ونعبر عن التقاطع بمايلي:

$$A \cap B = \{x | x \in A \wedge x \in B\}$$

مثال 3-10:

اذا كانت مجموعة $A = \{a, b, c, d\}$ ومجموعة $B = \{b, d, e\}$, فان تقاطع $A \cap B$ هو :

$$A \cap B = \{b, d\}$$

لاحظ ان العنصرين **b** و **d** هما العناصر المشتركة بين المجموعتين **A** و **B** فقط , وهو نتيجة التقاطع, اي ان **b** و **d** في مجموعة **A** وكذلك هم في مجموعة **B**.

تعريف 3-6 : اذا كانت **A** و **B** مجموعتين , فان اتحاد **A** و **B** يعبر عنه بدلالة :

$$A \cup B$$

ويقرأ **A اتحاد B** (A Union B) , ويعني مجموعة كل العناصر التي اما في **A** او اما في **B** . او في كلاهما ونعبر عن الاتحاد بمايلي:

$$A \cup B = \{x | x \in A \vee x \in B\}$$

لاحظ اننا استخدمنا **OR** الشاملة (inclusive or) , وهذا يعني ان x تنتمي لـ **A** , و x تنتمي لـ **B** , او ان x ربما في كل من **A** و **B**.

مثال 3-11:

اذا كانت $A = \{1,3,4,6,7\}$ و $B = \{3, 6, 7\}.$, فان اتحادهما $A \cup B$ هو :

$$A \cup B = \{1,3,4,6,7\}$$

مثال 3-12:

افرض ان $A = \{a, b, c, d, e\}$ و $B = \{a, c, e, f\}$, فاوجد $A \cup B$ و $A \cap B$

الحل:

ان $A \cap B$ هو مجموعة كل العناصر المشتركة الى كلا المجموعتين **A** و **B** وهي:

$$A \cap B = \{a, c, e\}$$

ان $A \cup B$ هو مجموعة كل العناصر في **A** او في **B** او في كلاهما وهي:

$$A \cup B = \{a, b, c, d, e, f\}$$

لاحظ ان **a** , **c** و **d** موجودة في كلا المجموعتين ونحن مازلنا نضع كل منهما مرة واحدة في المجموعة.

3-7-2 مكمُل المجموعة ***Complement of a Set***

تعريف 3-7 :

افرض ان **U** هي المجموعة الكونية Universal set , وافرض ان **A** هي مجموعة جزئية من **U** , فالمكمُل لـ **A** والذي يرمز له بـ A' ويقرأ (A Prime) او (A complement) هو مجموعة العناصر في **U** , لكن ليست في **A** , وهذا نعبر عنه بمايلي:

$$A' = \{x | x \in U \wedge x \notin A\}$$

هذه المجموعة نرمز لها ايضاً بـ $(U - A)$.

مثال 3-13:

اذا كانت **U** هي مجموعة الطلاب في المدرسة , و **A** هي مجموعة الطلاب الذين اخذوا مادة الجبر Algebra , فان A' هي مجموعة الطلاب الذين لم ياخذوا مادة الجبر . وبشكل مشابه , اذا كانت **U** هي مجموعة كل الحروف الابجدية الانكليزية , وكانت $A = \{s, l, y\}$, فان A' تمثل مجموعة كل الحروف الابجدية ماعدا الحروف s, l, y().

مثال3-14 :

افرض ان $U = \{1,2,3,4,5,6\}, A = \{1,3,5\}, B = \{2,4\}$, فاوجد

a. A`　　b.B`　　c. A` ⋂ B`　　d. A ⋂ B

الحل:

$a. A` = \{2,4,6\}$　$b. B` = \{1,3,5,6\}$　$c. A` \cap B = \{6\}$. اما الجواب **d** فهو $d. A \cap B = \phi$ وذلك بسبب عدم وجود عناصر مشتركة بين المجموعتين **A** و **B** .

لاحظ ايضاً بسبب ان $\phi \subseteq U$ و لايوجد عنصر في **U** هو عنصر في ϕ , فاننا نستنتج ان :

$$U' = \phi' \text{ و } \phi' = U$$

مثال 3-15:

اذا كانت $U = \{1,2,3,4,5,6\}, A = \{1,3,5\}, B = \{2,4\}$, فاوجد

$a.[A \cup B]'$ $\quad b.[A \cap B]'$ $\quad c.A' \cup B'$ $\quad d.A \cup [A \cup B]'$

الحل:

أ- $a. \quad A \cup B = \{1,3,5\} \cup \{2,4\} = \{1,2,3,4,5\}$

اذن $[A \cup B]' = \{6\}$, لاحظ اننا وجدنا اولاً $A \cup B$ ومن ثم وجدنا المتمم او المكمّل له (*it's complement*).

ب- $b. \quad A \cap B = \{1,3,5\} \cap \{2,4\} = \phi$

اذن: $[A \cap B]' = U = \{1,2,3,4,5,6\}$

ت-

$$A' \cup B' = \{1,3,5\}' \cup \{2,4\}' = \{2,4,6\} \cup \{1,3,5,6\} = \{1,2,3,4,5,6\}$$

ث- $A \cup [A \cup B]' = \{1,3,5\} \cup \{6\} = \{1,3,5,6\}$

لاحظ ان جواب الجزء **b** والجزء *c* متماثل , وان الجواب **c** من المثال الذي قبل هذا المثال متماثل مع جواب الجزء **a** من هذا المثال. ومن ذلك نستطيع ان نبين ان لأية مجموعتين **A** و **B** ان:

$$[A \cup B]' = A' \cap B' \text{ وان } [A \cap B]' = A' \cup B'$$

اخيرا يمكننا ان نوجد التقاطع والاتحاد والمكمُل لاكثر من مجموعتين كما في المثال التالي:

مثال 3-16 :

افرض ان لدينا المجموعات الاربعة التالية:

$$U = \{a, b, c, d, e, f\}, A = \{a, c, e\}, B = \{b, e\} \text{ and } C = \{a, b, d\},$$

فأوجد $[A \cup B] \cap C'$.

الحل:

بما ان $A \cup B$ هي مابين الاقواس , اذن نوجدها اولاً

$$A \cup B = \{a, c, e\} \cup \{b, e\} = \{a, b, c, e\}.$$

ثم نجد C' وهي :

$$C' = \{c, e, f\}$$

اذن :

$$[A \cup B] \cap C' = \{a, b, c, e\} \cap \{c, e, f\} = \{c, e\}$$

3-7-3 الفرق بين مجموعتين

Difference(or relative complement) of Two Sets

تعريف 3-8 : اذا كانت A و B مجموعتين , فان الفرق بين A و B يعبر عنه بدلالة :

$$A - B$$

ويقرأ A طرح B (A minus B) , ويعني مجموعة العناصر في A وليست في B . ونعبر عن الفرق بمايلي:

$$A - B = \{x | x \in A \wedge x \notin B\}$$

اما الفرق بين B و A فنعبر عنه بدلالة: $B - A$ وهو:

$$B - A = \{x | x \in B \wedge x \notin A\}$$

لاحظ ان تعريف A' هو حالة خاصة من التعريف اعلاه بسبب ان : $U - A = A'$, لاحظ ايضاً تعريف المجموعة المتممة او المكمُلة (*complement set*) السالف الذكر.

كما نلاحظ ان $(A \cap B' = A - B)$ وذلك بسبب ان $A \cap B'$ هي المجموعة لكل العناصر في **A** وليست في **B** وهذا بدقة هو تعريف ($A - B$).

مثال 3-17 :

افرض ان $A = \{1,2,3,4\}$, $B = \{1,2,5\}$, and $U = \{1,2,3,4,5,6\}$, فاوجد :

1. U - A 2. A` 3. A - B 4. B - A

-1 $(U - A)$: هي المجموعة لكل العناصر في **U** وليست في **A** . اي $(U - A = \{5,6\})$.

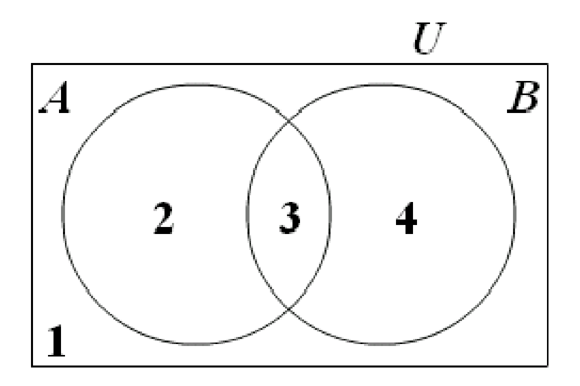
-2 A' : هي المجموعة لكل العناصر في **U** وليست في **A** . اي $(A' = \{5,6\})$.

-3 $(A - B)$: هي المجموعة لكل العناصر في **A** وليست في **B** . اي $(A - B = \{3,4\})$.

-4 $(B - A)$:هي المجموعة لكل العناصر في **B** وليست في **A** . اي $(B - A = \{5\})$.

3-7-4 مخطط فين Venn Diagram

ان هذا المخطط يزودنا بمنظر صوري للمجموعات . وفي مخطط فين, شكل (3-1) نمثل المجموعة الكونية بمستطيل (a *rectangle*) . والمجموعات الجزئية للمجموعة الكونية تمثل بدوائر(*circles*) :. وفي داخل الدائرة تمثل عناصر تلك المجموعة. انظر الشكل ادناه الذي نرى فيه مجموعتان **A** و **B** والمجموعة الكونية **U** . ان العناصر التي ليست في **A** وليست في **B** هي في الجزء (1). والجزء (2) يمثل العناصر في A وليست في B . الجزء (3) يمثل تقاطع A و **B** : $(A \cap B)$, والعناصر فيه مشتركة بين **A** و **B** . اما الجزء (4) فيمثل العناصرفي المجموعة **B** وليست في المجموعة **A** .



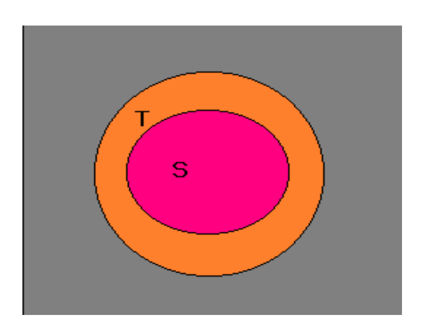
شكل (3-1) يبين مخطط فين Vein Diagram

ومثال اخر عندما نقول ان المجموعة **S** هي مجموعة جزئية من المجموعة **T** , فان ذلك يعني ان كل عنصر في **S** هو ايضا عنصراً في **T** ونعبر عن ذلك بمايلي:

$$S \subseteq T \Leftrightarrow \forall x\{x \in S \Rightarrow x \in T\}$$

حيث $\subseteq$ يمثل احتواء المجموعة (*Set Inclusion*) .

فعلى سبيل المثال اذا كانت $A = \{1,2,3,4,5\}$, فانها تعتبر مجموعة جزئية من $B = \{1,2,3,4,5,6\}$. . ويمكننا ان نمثل A بالصيغة التالية: $\{x \mid x \in I \wedge 0 < x < 6\}$ وفي هذه الحالة فان S تكون ضمن T . ويمكن تمثيلها بمخطط فين التالي شكل (3-2):



شكل (3-2) مخطط فين Vein Diagram

3-8 العلاقات بين المجموعات: Relationships Between Sets

نظرية 3-1 :

افرض ان **U** هي المجموعة الكونية (Universal set) وافرض ان **A** , **B** و C هي مجموعات جزئية من U فالخواص التي يمكننا الحصول عليها والتي تمثل العلاقات بين المجموعات هي مايلي :

1- قوانين الترابط **Associative Laws**

$$(A \cup B) \cup C = A \cup (B \cup C), \qquad (A \cap B) \cap C = A \cap (B \cap C)$$

2- قوانين التبادل **Commutative Laws**

$$A \cup B = B \cup A, \qquad A \cap B = B \cap A$$

3- فوانين التوزيع **Distributive Laws**

$$(A \cap (B \cup C) = (A \cap B) \cup (A \cap C),$$
$$(A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$$

4- قوانين التماثل **Identity Laws**

$$(A \cup \phi = A, \qquad A \cap U = A$$

5- قوانين التكامل **Complement Laws**

$$A \cup \overline{A} = U \qquad A \cap U = \phi$$

6- قوانين **Idempotent Laws**

$$A \cup A = A \qquad A \cap A = A$$

7- قوانين الربط **Bound Laws** :

$$A \cup U = U \qquad A \cap \phi = \phi$$

8- قوانين الانهماك **Absorption Laws**

$$A \cup (A \cap B) = A \qquad A \cap (A \cup B) = A$$

9- قانون **Involution Law**

$$\overline{\overline{A}} = A$$

10- قوانين الصفر/واحد **0/1 Laws** :

$$\overline{\phi} = U \qquad \overline{U} = \phi$$

11- قوانين دي موركان للمجموعات **De Morgan;S Laws for Set**

$$\overline{(A \cup B)} = \overline{A} \cap \overline{B}, \qquad \overline{(A \cap B)} = \overline{A} \cup \overline{B}$$

نظرية 3-2

افرض ان **R** و **S** مجموعتان فان:

1- **R** و **S** هما مجموعتان جزئيتان من المجموعة الناتجة من اتحاد **R و S** :

R and S *are subsets of* $R \bigcup S$.

2- تقاطع **R** و **S** هو مجموعة جزئية من كل من **R** و **S** :

$R \bigcap S$ *is a subset of both* R & S

3- اتحاد **R** و **S** يساوي **S** اذا وفقط اذا **R** هي مجموعة جزئية من **S** :

$R \bigcup S = S$ *if and only if* $R \subseteq S$.

4- تقاطع **R** و **S** يساوي **R** اذا وفقط اذا **R** هي مجموعة جزئية من **S** :

$R \bigcap S = R$ *if and only if* $R \subseteq S$

ومن العلاقات المهمة بين عمليات التكملة (*complementation*) , الاتحاد (*union*) , و التقاطع (*intersection*) هي قوانين دي موركان للمجموعات(**De Morgan;S Laws for Set**)

$$\overline{(A \bigcup B)} = \overline{A} \bigcap \overline{B}, \qquad \overline{(A \bigcap B)} = \overline{A} \bigcup \overline{B}$$

وهي نفسها يمكن التعبير عنها بالصيغة التالية:

$$\neg(A \bigcup B) = \neg A \bigcap \neg B, \qquad \neg(A \bigcap B) = \neg A \bigcup \neg B$$

كما ذكرنا سابقاً.والتي يمكن برهنتها باستخدجام مخطط فين Vein Diagram .

نحن نُعرف اتحاد (Union)عائلة عشوائية **S** من المجموعات على انها تلك العناصر x التي تنتمي على الاقل الى مجموعة واحدة X في S . ونكتبها بالصيغة التالية:

$$\bigcup S = \{x \mid x \in X \quad for \quad some \quad X \in S\}$$

وبشكل مشابه نعرف تقاطع (*intersection*) عائلة عشوائية S من المجموعات على انها تلك العناصر x التي تنتمي لأية مجموعة **X** في **S** . ونكتبها بالصيغة التالية:

$$\bigcap S = \{x \mid x \in X \quad for \quad all \quad X \in S\}$$

فاذا كانت **S** هي:

$$S = \{A_1, A_2, ...A_n\}$$

فاننا نكتب الاتحاد والتقاطع لعائلات **S** كمايلي:

$$\bigcup S = \bigcup_{i=1}^{n} A_i \,, \qquad \bigcap S = \bigcap_{i=1}^{n} A_i$$

لكن اذا كانت **S** هي:

$$S = \{A_1, A_2, ...\}$$

فاننا نكتب الاتحاد والتقاطع لعائلات **S** كمايلي:

$$\bigcup S = \bigcup_{i=1}^{\infty} A_i \,, \qquad \bigcap S = \bigcap_{i=1}^{\infty} A_i$$

مثال 3-18

اذا كانت

$A_n = \{n, n+1, ...\}$ و $S = \{A_1, A_2, ...\}$ فان:

$$\bigcup_{i=1}^{\infty} A_i = \bigcup S = \{1,2,...\}$$

وان

$$\bigcap_{i=1}^{\infty} A_i = \bigcap S = \phi$$

من جانب اخر ان جزء من المجموعة **S** يقسم **X** الى مجموعات جزئية غير متداخلة .واكثر صياغة ,نعني بتجمع **S** لمجموعات جزئية غير فارغة لـ **X** يقال انه جزء *(a partition)* من المجموعة **X** اذا كل عنصر في **X** ينتمي بالضبط الى عنصر واحد من **S** . لاحظ انه اذا **S** هي جزء من **X** , فان **S** هي *(pairwise disjoint)* وان $\bigcup S = X$.

مثال 3-19:

بما ان كل عنصر من **X** : $X = \{1,2,3,4,5,6,7,8\}$ هو بالضبط عنصر واحد من **S** :

$$S = \{\{1,,4,5\}, \{2,6\}, \{3\}, \{7,8\}\}$$

فان **S** هي جزء *(partition)* من **X** .

لقد ذكرنا سابقاً ان المجموعة هي مجموعة من العناصر ليس بالضرورة مرتبة , بمعنى اخر ان المجموعة تحدد بعناصرها وليست باي ترتيب خاص لعناصرها . لكن في بعض الاحيان نود اخذ الترتيب بالحسبان . فاي زوج مرتب نكتبه بـ **(a, b)** هو مختلف عن الزوج **(b, a)** الاُ الهمً **(a = b)** . ولنضع ذلك بمعنى اخران :

$$(a,b) = (c,d) \quad \underline{\text{if and only if}} \quad a = c, \text{ and } b = d$$

فاذا كانت **X** و **Y** مجموعتان , نحن نفرض XxY ترمز الى مجموعة كل الازواج المرتبة (x,y) , حيث x تنتمي الى **X** وان **y** تنتمي الى **Y** . ونسمي XxY الضرب الكارتيزي *(Cartesian product)* لـ **X** و **Y**.

مثال 3-20 :

اذا كانت $X = \{1,2,3\}$, و $Y = \{a,b\}$ فأن :

$$XxY = \{(1,a),(1,b),(2,a),(2,b),(3,a),(3,b)\}$$

$$YxX = \{(a,1),(b,1),(a,2),(b,2),(a,3),(b,3)\}$$

$$XxX = \{(1,1),(1,2),(1,3),(2,1),(2,2),(2,3),(3,1),(3,2),(3,3)\}$$

$$YxY = \{(a,a),(a,b),(b,a),(b,b)\}$$

وبشكل عام من هذا المثال نرى ان :

$$XxY \neq YxX$$

$$|XxY| = |X| \ . \ |Y|$$

مثال 3-21:

اذا كانت $X = \{1,2\}$, $Y = \{a,b\}$ و $Z = \{\alpha,\beta\}$ فأن :

$$XxYxZ = \{(1,a,\alpha),(1,a,\beta),(1,b,\alpha),(1,b,\beta),(2,a,\alpha),(2,a,\beta),(2,b,\alpha),(2,b,\beta)\}$$

$$|XxYxZ| = |X| \ . \ |Y| \ . \ |Z|$$

وبشكل عام :

$$|X_1 x X_2 x ....x X_n| = |X_1| \ . \ |X_2| \ .... \ |X_n|$$

9-3 عمليات المجموعة والادوات المنطقية Set Operations & logical connectives

الرموز التي استخدمناها لعمليات المجموعة (*Set Operations*) هي $(\cup, \ \cap, \ \sim)$ والتي هي مشابهة الى الرموز $(\vee, \ \wedge, \ \neg)$ التي استخدمناها في الادوات المنطقية (*logical connectives*) التي تستخدم مع الجمل.

نظرية 3-3

افرض ان **p** و **q** هما جملتان (*two statements*) وافرض ان **Q, P** هما المجموعتان الحقيقيتان (*Truth Sets*) لهما , فان:

1- $P \cap Q$ هي المجموعة الحقيقية لـ $p \wedge q$.

2- $P \cup Q$ هي المجموعة الحقيقية لـ $p \vee q$.

3- ~P هي المجموعة الحقيقية لـ $\neg p$

ونرى في هذه النظرية اننا غيرنا كل الحروف الصغيرة (*Small letters*) التي تشير الى الجمل (*statements*) الى حروف كبيرة (*Capital letters*) التي تشير الى المجموعات (Sets). كما غيرنا كل رموز الربط الزاوية (نسبة الى شكل الزاوية) $(\vee, \ \wedge, \ \neg)$ الى رموز ربط منحنية $(\cup, \ \cap, \ \sim)$.وهذا يقودنا الى حقائق تكافئ الجمل (*statement equivalent*) التي مثل :

$$p \wedge (q \vee r) \Leftrightarrow (p \wedge q) \vee (p \wedge r)$$

وفي هذه الحالة يمكننا استخدام مخطط فين لتمثيل ذلك التكافؤ بالاعتماد على ماورد في النظرية اعلاه.وكما سنرى في التعريف التالي:

تعريف 3-9 :

اذا كانت **p , q** و **r** تقوم مقام الجمل (*statements*) حول المتغير **x** . فانه يوجد مجموعة حقيقية **P** للجملة **p** , ومجموعة حقيقية **Q** للجملة **q** , واخيرا مجموعة حقيقية **R** للجملة **r**. اذن بواسطة النظرية (3-2) اعلاه , فان :

1- المجموعة الحقيقية للجملة المركبة $p \wedge (q \vee r)$ هي $P \bigcap (Q \bigcup R)$.

2- المجموعة الحقيقية للجملة المركبة $(p \wedge q) \vee (p \wedge r)$ هي $(P \bigcap Q) \bigcup (P \bigcap R)$.

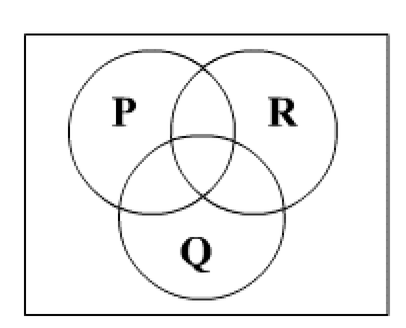
لكن بواسطة تعريف التكافئ اعلاه ($(p \wedge (q \vee r) \Leftrightarrow (p \wedge q) \vee (p \wedge r))$) نرى ان $p \wedge (q \vee r)$ مكافئة لـ $(p \wedge q) \vee (p \wedge r)$ اذا وفقط اذا كانت $P \bigcap (Q \bigcup R)$ مساوية الى $(P \bigcap Q) \bigcup (P \bigcap R)$, وهذا مايشير له قانون التوزيع (*Distributed law*) الانف الذكر .وهو مانستطيع اثباته بواسط مخطط فين. وهذه الحقائق يمكننا ان نطلق عليها قانون الربط بين الجمل والمجموعات نتيجة ان لكل جملة هناك مجموعة حقيقة لها سواء كانت الجملة بسيطة او مركبة كما ذكرناه ضمن هذا التعريف (3-9).

مثال 3-22:

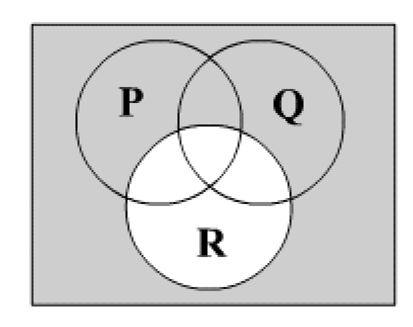
ارسم مخطط فين (***Venn Diagram***) وظلل المنطقة الممثلة بواسطة $P \bigcup (Q \bigcap \sim R)$.

الحل:

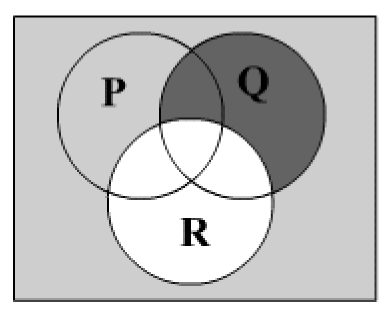
بما أنه لدينا العلاقة $P \bigcup (Q \bigcap \sim R)$, اذن ستكون لدينا المجموعات **P ,R ,Q** , وبالتالي ستكون في الشكل(3-3), ثلاثة دوائر متقاطعة متداخلة كما هو في الجزء (**a**) , وان نجد ~R وهو المكمل لـ **R** كما في الجزء (**b**) , ومن ثم نجد $Q \bigcap \sim R$ (الجزء (c)) , واخيرا نجد اتحاد **P** مع ($Q \bigcap \sim R$) وهو في الجزء (**d**).



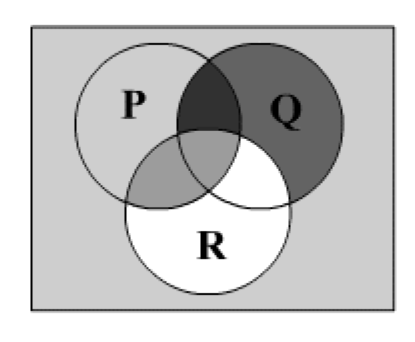
(a)ثلاثة دوائر تمثل ثلاث مجموعات



(b) يمثل المجموعة المكملة لـ R (~R)




(c) يمثل $Q \cap \sim R$



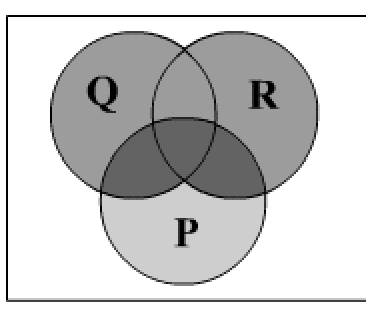
(d) يمثل $P \cup (Q \cap \sim R)$

شكل(3-3) يمثل مخطط فين (Venn Diagram) للمثال (22-3)

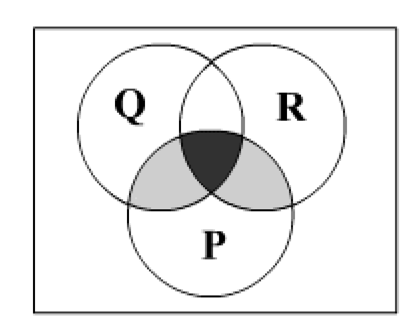
مثال 3-23:

برهن بواسطة مخطط فين ان $P \cap (Q \cup R)$ تساوي $(P \cap Q) \cup (P \cup R)$.

Prove by Venn Diagram that $P \cap (Q \cup R)$ *equals* $(P \cap Q) \cup (P \cup R)$



$P \cap (Q \cup R)$



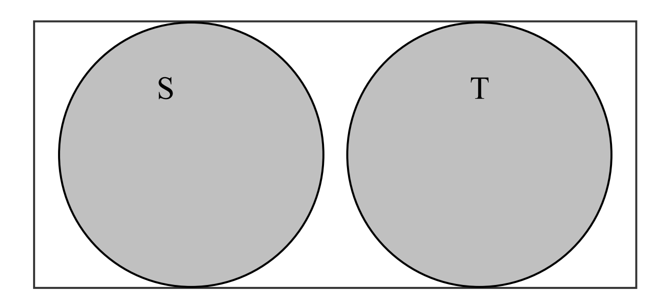
$(P \cap Q) \cup (P \cup R)$

شكل(3-4) يمثل مخطط فين (Venn Diagram) للمثال (23-3)

مثال 3-24:

بين مخطط فين لمجموعتين منفصلتين S و T.:

Show Venn Diagram of two disjoint sets



شكل (3-5)بين مخطط فين لمجموعتين منفصلتين S وT

مـن الشـكل المجـاور نـرى ان اتحـاد المجمـوعتين يسـاوي ϕ , بمعنـى انهـما منفصـلان. كـما ان لكـل مجموعـة نجـد ان $S \cup \phi = S$ و $S \cap \phi = \phi$. اذن من نظرية 1 نجد ان $S \subseteq \phi$.ومن نظرية 3- 1 , فان:

$S \cap T = \phi$ (*null or* ϕ). *for every set* $S \cup \phi = S$ *and* $S \cap \phi = \phi$, $\phi \subseteq S$)

3-10 العلاقة بين المجموعات المختلفة للاعداد The relationship of the various sets of numbers

تتمثل تلك العلاقة بالشكل(3-6) التالي :

C , The complex numbers

R, The real numbers

Q , The rational numbers

I , The integer numbers

W , The whole numbers

N , The counting numbers

N = {1, 2, 3, ...}

W = { 0, 1, 2, 3, ...}

I = { .....,-2, -1, 0, 1, 2, 3, ...}

Numbers of the form a/b ,where a and b are integers and $b \neq 0$

All the rational numbers and all irrational numbers

numbers of the form a + bi, where a and b are real numbers and $i = \sqrt{-1}$

شكل(3-6) يبين العلاقة بين المجموعات المختلفة للاعداد

ومن خلال الشكل نجد ان العلاقة هي:

$$N \subset W \subset I \subset Q \subset R \subset C$$

11-3 المتتالية The Sequence

ان الدالة التي مجالها (*domain*) هو مجموعة من الاعداد الصحيحة المتتالية نسميها بالمتتالية (*sequence*) . فعلى سبيل المثال الدالة $S(i)$ لكل عدد صحيح (**i**) اكبر من صفر :

$$S(i) = i^2 , \text{for each integer} \quad i \geq 0$$

تسمى متتالية . ولاسباب تأريخية وللملائمة تم استخدام رمزخاص للدلالة على المتتالية غير الرمز $S(i)$, وقد استخدم S_i , وبشكل نموذجي اننا نشير الى S_i بانه الحد i^{th} للمتتالية ونشير الى $< S_i >$ لتقوم مقام المتتالية, ومن جانب اخر اذا رمزنا للمتتالية بـ $\{S_n\}$ فان ذلك يشير الى المتتالية الكاملة (*entire sequence*) , اي يشير الى:

$$S_1, \quad S_2 ,...... \quad S_n ,......$$

كما يمكن القول ان المتتالية هي عبارة عن قائمة ياخذ الترتيب فيها بنظر الاعتبار , فمثلاً القائمة المرتبة (*ordered list*) التالية:

$$2, 4, 6 ,.........2n,.....$$

هي عبارة عن متتالية (*sequence*) , العنصر الاول فيها هو 2 والعنصر الثاني هو 4 والعنصر n هو 2n وهكذا. اذا فرضنا ان S يرمز الى تلك المتتالية فان:

$$S_1 = 2 \qquad S_2 = 4,...... \quad S_n = 2n,......$$

مثال 3-25:

القائمة المرتبة التالية :

$$a, a, b , a, b$$

هي عبارة عن متتالية (*sequence*) , العنصر الاول فيها هو a والعنصر الثاني هو a , وهكذا. فاذا فرضنا ان **t** يرمز الى تلك المتتالية فان:

$$t_1 = a \quad t_2 = a \quad ,t_3 = b..... \quad t_5 = b,......$$

ان هذا المثال يبين ان المتتالية تتكرر فيها العناصر, وهذا عكس المجموعة (Set) التي لاتتكرر فيها العناصر

مثال 3-26:

عرِّف المتتالية $\{t_n\}$ حسب القانون التالي:

$$t_n = n^2 - 1, \qquad n \geq 1$$

اذن ستكون العناصر الخمسة الاولى كمايلي:

$$0, \ 3, \ 8, \ 15, \ 24$$

اما الحد رقم **55** فهو:

$$t_{55} = 55^2 - 1 = 3024$$

مثال 3-27:

اكتب الحد الثالث للمتتالية $< S_i >$ المعطاة بالشكل التالي:

$$S_i = i(i-1)+1 \qquad i \geq 1$$

الحل:

بما انه المطلوب الحد الثالث . اذن (i) قيمتها تساوي 3 $(i=3)$. فسيكون الحد الثالث هو:

$$S_i = 3(3-1)+1 = 7$$

ان المتتالية السابقة تعتبر متتالية غير محددة (*infinite sequence*) لان حدودها غير محدودة . هذا من جانب , لكن من جانب اخر انه احياناً مجال الدالة لايحتاج ان يكون غير محدد , اذ اننا في الغالب يكون لدينا مجال الدالة عبارة عن المجموعة التالية:

$$N = \{1,2,....,n\}$$

وفي حالة الاحداثيات (الازواج المرتبة) (*ordered pair*) فاننا نفكر بمتتالية مجالها هو مثلاً $\{1,2\}$,او مرتب بشكل ثلاثي (*ordered triple*) مثل $\{1,2,3\}$ وهكذا. اما المتتالية التي مجالها مثل :

$$\{1,2,3,.......n\}$$

تدعى متعدد الترتيب (*ordered n-tuple*) .

ان المتتالية المحددة تسمى *string* : *a finite sequence is called string*

مثال 3-28:

اذا كان : $S = (\text{Sam, Ali, Ahmed, William})$ يمثل متتالية محددة من الاسماء فانه يدعى **4-tuple** , وترتيب حدوده هو: S_1=Sam .و S_4=William , وهكذا. ان الترتيب هنا ضروري ومهم عكس الحالة في المجموعات (Sets) حيث ترتيب العناصر غير ضروري كما اشرنا سابقاً.

مثال 3-29:

اذا كانت **x** هي المتتالية المعرّفة كمايلي:

$$x_n = \frac{1}{2^n} \qquad -1 \le n \le 4,$$

فان عناصر **x** هي:

$$2, \ 1, \ 1/2, \ 1/4, \ 1/8, \ 1/16$$

هناك نوعين مهمين من المتتاليات , متتاليات متزايدة (*increasing sequences*) ومتتاليات متناقصة (*decreasing sequences*) . فمثلاً المتتالية s هي متزايدة اذا كان: [1]

$$s_n \le s_{n+1} \qquad \text{for all } n$$

وتعتبر متناقصة اذا كان:

$$s_n \ge s_{n+1} \qquad \text{for all } n$$

مثال 3-30:

المتتالية: $2, 4, 6....,2n$ هي متتالية متزايدة بسبب ان :

$$s_n = 2n \le 2(n+1) = s_{n+1} \qquad \text{for all } n$$

اما المتتالية: $2, 1, 1/2, 1/4, 1/8, 1/16$ هي متتالية متناقصة بسبب ان:

$$x_n = \frac{1}{2^n} \ge \frac{1}{2^{n+1}} = x_{n+1} \qquad \text{for all } n$$

تعريف 3-10:

افـرض ان $\{s_n\}$ هـو متتاليـة معرَّفـة لــ $(n = m, m+1,....)$ وافـرض ان $(n_1, n_2....)$ هـي متتالية متزايدة تحقـق العلاقـة $(n_k < n_{k+1} \quad \text{for all} \quad k)$. لكـل قـيم k التـي قيمهـا هـي المجموعـة $\{m, m+1,.....\}$. فاننا ندعو المتتالية $\{s_{n_k}\}$ هي متتالية جزئية *(subsequence)* من $\{s_n\}$.[1]

مثال 3-31:

ان المتتالية : $2, 4, 8, 16.........2^k,.....$ (1)

هي متتالية جزئية من المتتالية التالية: $2, 4, ,6, 8, 10, 12, 14, 16.........2n,.....$ (2)

ان المتتالية الجزئية (1) تحصل من المتتالية الثانية(2) بواسطة اختيار الحدود الاول والثـاني والرابـع والثـامن منهـا (اي مـن المتتالية 2) وهكذا. وبالتالي فان قيمة n_k حسب التعريف اعلاه هي:

$$n_k = 2^{k-1}$$

فلو عرَّفنا المتتالية (2) بواسطة $s_n = 2n$, فان المتتالية الجزئية تعرَّف بـ

$$s_{n_k} = s_{2^{k-1}} = 2.2^{k-1} = 2^k$$

من جانب اخر ان هناك طريقتان لانشاء متتاليات جديدة من متتاليات عددية وهما بواسطة عمليتي اضافة وضرب الحدود سوية كما في المثال التالي:

مثال 3-32:

اذا كان $\{a_i\}_{i=m}^{n}$ هو متتالية : فاننا نعرَّف :

$$\sum_{i=m}^{n} a_i = a_m + a_{m+1} + ... + a_n$$

ونعرَّف :

$$\prod_{i=m}^{n} a_i = a_m . a_{m+1} .....a_n$$

ان الصيغة $\sum_{i=m}^{n} a_i$ تسمى رموز الجمع او (*sum* (*or sigma*) *notation*) . والصيغة $\prod_{i=m}^{n} a_i$ تسمى رموز الضرب (*product notation*) . كما ان(**i**) تسمى المعامل (*index*) وان (**m**) تسمى الحد الادنى (*lower limit*) , و (**n**) تمثل الحد الاعلى (*upper limit*).

مثال 3-33:

افرض ان **a** هي متتالية معرَّفة بـ ($a_n = 2n \quad , n \geq 1$) , فان المجموع والضرب كمايلي:

$$\sum_{i=1}^{3} a_i = a_1 + a_2 + a_3 = 2 + 4 + 6 = 12$$

$$\prod_{i=1}^{3} a_i = a_1.a_2.a_3 = 2.4.6 = 48$$

مثال 3-34:

ان المجموع الهندسي التالي:

$$a + ar + ar^2 + \ldots + ar^n$$

يمكننا كتابته بدلالة رمز المجموع (Sum Notation)كمايلي

$$\sum_{i=0}^{n} ar^i$$

ان اسم المعامل (*the index*)في المثالين السابقين هو غير متصل بالموضوع (irrelevant). فعلى سبيل المثال:

$$\sum_{i=0}^{n} a_i = \sum_{j=0}^{n} a_j$$

وكذلك

$$\prod_{i=1}^{n} a_i = \coprod_{j=1}^{n} a_j$$

ان ذلك احياناً مهم ليس فقط لتغير اسم المعامل من i الى j وانما لتغيير حدود الجمع .ان العملية هي مشابهة (*analogous*) لتغيير المتغير في التكامل (*integral*) .

3-11-1 تغيير المعامل والحدود بالمجموع *Changing The Index And Limits In Sum*

مثال 3-35:

اعد كتابة المجموع :

$$\sum_{i=0}^{n} ir^{n-i}$$

مبدِّلاً **i** بـ **j** حيث $(i = j-1)$

الحل:

بما ان $(i = j-1)$, اذن الحد (ir^{n-i}) يصبح

$$(j-1)r^{n-(j-1)} = (j-1)r^{n-j+1})$$

وبما ان $(j = i+1)$ من منطوق السؤال. اذن عندما $(i = 0)$ فان $(j = 1)$, وهذا يعني ان الحد الادنى لـ **j** هو **1** .
وبشكل مشابه عندما $(i = n)$, فان $(j = n+1)$. وبذلك يكون الحد الاعلى لـ **j** هو $(n+1)$, وعلى اساس ذلك فان:

$$\sum_{i=0}^{n} ir^{n-i} = \sum_{j=1}^{n+1} (j-1)r^{n-j+1}$$

مثال 3-36:

افرض ان **a** هي متتالية معرّفة بالمعادلة التالية: $(a_n = 2(-1)^n, \quad n \geq 0)$, اوجد صيغة للمتتاليـة **s** المعرّفـة بواسطة المعادلة التالية:

$$s_n = \sum_{i=0}^{n} a_i$$

فاننا نجد ان:

$$\begin{aligned} s_n &= 2(-1)^0 + 2(-1)^1 + 2(-1)^2 + \ldots + 2(-1)^n \\ &= 2 - 2 + 2 - \ldots.. \pm 2 = \begin{cases} 2 & \text{if } n \text{ is even} \\ 0 & \text{if } n \text{ is odd} \end{cases} \end{aligned}$$

احياناً رموز الجمع والضرب تعدّل لترمز الى الجمع والضرب المفهرس على مجموعات من الاعداد الصحيحة . وبشكل صـيغة , اذا كانت **S** هي مجموعة (**Set**) من الاعداد الصحيحة و **a** هي متتالية فان :

$$\sum_{i \in S} a_i$$

تدل على المجموع للعناصر $\{a_i \mid i \in S\}$

وبشكل مشابه فان :

$$\prod_{i \in S} a_i$$

تدل على الضرب للعناصر $\{a_i \mid i \in S\}$

مثال 3-37 :

اذا كانت **S** ترمز الى مجموعة الاعداد الاولية اقل من **20** فان المجموع $\sum_{i \in S} \frac{1}{i}$ هو:

$$\sum_{i \in S} \frac{1}{i} = \frac{1}{2} + \frac{1}{3} + \frac{1}{5} + \frac{1}{7} + \frac{1}{11} + \frac{1}{13} + \frac{1}{17} + \frac{1}{19} = 1.455$$

ان المتتالية المحددة تدعى ***string*** .

تعريف 3-11: ان مجموعة حروف string على X هو متتالية محددة لعناصر من X

A string over X is a sequence of elements from X .[1]

مثال 3-38 :

افرض ان $X = \{a, b, c\}$. فاذا فرضنا ان:

$$\beta_1 = b, \quad \beta_2 = a, \quad \beta_3 = a, \quad \beta_4 = c$$

فاننا نحصل على (*string*) على X , يكتب بـ (baac)

وبما ان الـ (*string*) هو متتالية , اذن الترتيب order ياخذ بنظر الاعتبار . فعلى سبيل المثال مجموعة الحـروف (*string*), " baac " تختلف عن " acba ".

ان التكرار (*repletion*) في الـ (*string*) يمكن وصفه بدلالة اس يساوي تكرار الحرف (*superscripts*) , فعلى سبيل المثال bbaaac يمكن كتابتها بـ b^2a^3c

ان الـ (*string*) الخالي من العناصر يسمى (*null string*) ويرمز له بـ λ.

وترمز X^* الى مجموعة كل الـ (*Strings*) على **X** ومن ضمنها الـ (*Null string*) . كما اننا نفرض ان X^+ , وترمز الى مجموعة كل الـ (*null strings*) على **X** .

مثال 3-39:

افرض ان $X = \{a, b\}$. بعض العناصر في X^* هي :

$$\lambda, \quad a, \quad b, \quad abab, \quad b^{20}a^5ba$$

ان طول الـ (*string*) , " α " هو عدد العناصر في " α " , وطول " α " يرمز له بـ " $|\alpha|$ "

مثال:

اذا كانت $(\alpha = aabaa)$ و $(\beta = a^3b^4a^{32})$ فان:

$$|\alpha| = 5, \qquad |\beta| = 39$$

فاذا كانت α و β هما (*string*) فان الـ (*string*) الذي يشتمل على α متبوعاً بـ β يكتب $\alpha\beta$ ويدعى بـ التسلسل او التتابع (*concatenation*) لـ α و β .

مثال 3-40:

اذا كانت $\theta = \text{cabd}$ $\quad \gamma = \text{aab},$ فان :

$$\lambda\gamma = \gamma = \text{aab} \qquad \gamma\lambda = \gamma = \text{aab} \qquad \theta\gamma = \text{cabdaab} \qquad \gamma\theta = \text{aabcabd}$$

3-11-2 جمع المتتاليات المحدودة: Summing Finite Sequences

1-اذا كنا نريد معرفة الزمن اللازم لتنفيذ عملية مكونة من عدة خطوات(مراحل)

2- ونريد معرفة كمية الإنتاج المحصورة من اليوم العاشر مثلاً الى يوم ما اخر. فلو كانت a_i تقوم مقام كمية الانتاج لمصنع في اليوم (**i**). فكمية الانتاج خلال اليوم **m** الى اليوم **n** هي مجموع القيم a_i لكل قيم (i) المحصورة بين **m** و **n** .وكمايلي:

$$\sum_{i=m}^{n} a_i = \text{a}_\text{m} + \text{a}_{\text{m}+1} + \text{a}_{\text{m}+2} + \ldots\ldots.\text{a}_\text{n}$$ بدلالة معادلة

وبدلالة الكلمات (*by words*), فان $\sum_{i=m}^{n} a_i$ تقوم مقام نتيجة الحساب a_i لكل (i) بين **m** و **n** ومن ثم اضافة تلك القيم الى بعضها الاخر للحصول على النتيجة النهائية .

مثال 3-41 :

اوجد قيمة $\sum_{i=1}^{3} i^2$. و قيمة $\sum_{i=2}^{5} 3^i$

الحل:

1- $\sum_{i=1}^{3} i^2 == 1^2 + 2^2 + 3^2 = 1 + 4 + 9 = 14$

2- $\sum_{i=2}^{5} 3^i == 3^2 + 3^3 + 3^4 + 3^5 = 360$

ان المجموع (SUM) المهم لنا هو المجموع التالي:

$$1 + 2 + 3 + \ldots\ldots\ldots + n = \sum_{i=1}^{n} i \quad \text{OR} \quad \sum_{i=1}^{n} i = n(n+1)/2 \quad \textit{Gauss's formula}$$

3-11-3 المتسلسلات الحسابية Arithmetic series

واحدة من الامور المهمة لنا باستخدام صيغة كاوس (*Guass's formula*) **هي حساب المجموع للصيغة التالية :**

$$\sum_{i=1}^{n} (ai + b)$$

الذي يسمى المتسلسلة الحسابية (Arithmetic series) , حيث a و b ثوابت.

نظرية 3-4 :

اذا كان $< a_n >$ و $< b_n >$ هما متتاليتان (sequences) وكانت c ثابت فان:

Theorem1 *if* $< a_n >$ *and* $< b_n >$ *are sequences and c is a constant , then*

$$\sum_{i=m}^{n} (a_i + b_i) \equiv \sum_{i=m}^{n} a_i + \sum_{i=m}^{n} b_i$$

$$\sum_{i=m}^{n} ca_i = c\sum_{i=m}^{n} a_i \quad and \quad \sum_{i=m}^{n} c = (n - m + 1)c.$$

نظرية 3-5 :

مجموع المتسلسلة الحسابية هو:

$$a.n(n+1)/2 + ab$$

البرهان:

من صيغة كاوس اعلاه نحصل على :

$$\sum_{i=1}^{n} i = n(n+1)/2$$

ثم بالاعتماد على **النظرية 1** اعلاه نحصل على مايلي:

$$\sum_{i=1}^{n}(ai+b) = a\sum_{i=1}^{n} i + \sum_{i=1}^{n} b \quad = an(n+1)/2 \quad + bn.$$

مثال 3-42:

اوجد المجموع $\sum_{i=1}^{10}(3i-1)$

لايجاد المجموع نكتب مايلي:

$$\sum_{i=1}^{10}(3i-1) = 3*10(11)/2 \quad +10*(-1) = 165-10 = 155$$

او باستخدام التعريف (نجد قيمة كل حد عند كل قيمة لـ (i) ثم نجمع الكل) :

$$\sum_{i=1}^{10}(3i-1) = 2+5+8+11+14+17+20+23+26+29 = 155$$

3-12 الدوال والرسوم البيانية Functions And Graphs

3-12-1 العلاقات والدوال Relations And Functions

في الرياضيات نحن غالباً ما نـدعو العـدد الاول في الـزوج المُرتَّـب (*ordered pair*) بــقيمة x والعـدد الثـاني بــقيمة y , ويرمز لهذا الزوج بـ (x,y) والمعادلة :

$$(x,y) = (2,-5)$$

تعني ان $(x=2)$ و $(y=-5)$ وبشكل عام فان :

$$(x,y) = (a,b) \quad \text{if and only if} \quad x=a, y=b$$

وتوجد مسائل كثيرة نحتاج فيها الى دراسة مجموعات الازواج المُرتَّبة , وبشكل مناسب ندعو مثل تلك المجموعة بالعلاقة(*relation*) . ان وجود الزوج المرتب **(a, b)** في علاقة رياضية يفسر كمؤشر على وجود علاقة (*relationship*) من **a** الى **b** .

ان نموذج قواعد البيانات العلائقية (*Relational Database Model*) , التي تساعد المستخدمين في الدخول الى البيانات في قاعدة البيانات (التي هي مجموعة من السجلات (*records*) المعالجة بواسطة الحاسوب) أُسِّست على اساس مبدأ العلاقة (*relation*). كما ان فكرة الزوج المُرتَّب تستخدم في عدة مجالات منها على سبيل المثال تحديد احداثيات الموقع *location coordinates*

تعريف 3-12:

العلاقة هي مجموعة من الازواج المُرتَّبة A relation is a set of ordered pairs

مثال 3-43:

ان المجموعة التالية:

$$R = \{(4,-3),(2,-5),(-3,4)\}$$

هي علاقة تم فيها تحديد كل الازواج المُرتَّبة , حيث مجموعة العناصر الاولى تمثل ***x*** هي $\{-3,2,4\}$ ومجموعة العناصر الثانية تمثل ***y*** هي $\{-5,-3,4\}$, وان ادراج العناصر في العلاقة **R** يبين كيف ان عناصر المجموعة الاولى مرتبطة بعناصر المجموعة الثانية لتشكيل الازواج المُرتَّبة للعلاقة المعطاة.ان طرق مختلفة في ارتباط عناصر المجموعتين يعطي علاقة مختلفة , فمثلاً:

$$S = \{(-3,-5),(2,-3),(4,4)\}$$

هي علاقة مختلفة عما هو لـ **R** رغم ان العلاقتين تشكلت من نفس المجموعتين الانفة الذكر , وذلك لانه على سبيل المثال ان الزوج المُرتَّب : $(4,-3) \neq (-3,4)$

3-12-2 فكرة العلاقة : The idea of the relation

تصوًر ان لدينا الصيغتين التاليتين :

$$y = x + 2, \qquad and \quad y = x^4 - 2x^3 - x^2 + 3x + 2$$

فاننا نلاحظ ان كلا الصيغتين لهما مجموعة حقيقية (*truth set*) وكلاهما يحوي متغيرين (Two variables).

(x و y) , اذن نحتاج ان نعطي قيمتين , واحدة لـ x والثانية لـ y قبـل ان نقـرر صـحة او خطـأ الجملـة . ولـكي نكتـب المجموعة الحقيقية لكل جملة (صيغة) , فاننا نحتاج الى طريقة لنسجل اي قيمة لـ x تقابل اي قيمة لـ y , و نكتـب تلـك القيم بشكل (x,y) حيث x تمثـل القيمـة الاولى , ويـدعى (x,y) بـالزوج المُرتَّـب (*ordered pair*) بسـبب وجـود عددين وبسبب ان الترتيب مهم جدا .

من جانب اخر ان كلا الصيغتين يصفان العلاقة بين المجموعة $\{2,1,0,-1\}$ التي تمثل المجموعة الكونية لـ x والمجموعة $\{4,3,2,1\}$ التي تمثل المجموعة الكونية لـ y . ان معنى ذلك ان كلا الصيغتين يربط (-1) الى (1) , 0 الى 2, 1 الى 3 , و 2 الى 4 . وعلى اساس ذلك ان المجموعتين الكونيتين المعطاة يربطان بالضبط نفس الزوج المُرتَب , أي بتعبير اخر ان الصيغتين هما نفس الصيغتين.

ان المجموعة الحقيقية لـ $y = x^4 - 2x^3 - x^2 + 3x + 2$ و $y = x + 2,$ هي مجموعة الازواج المُرتَّبة التالية : $\{(-1,1),(0,2),(1,3),(2,4)\}$.

من جانب اخر وفي معظم الحالات فاننا مهتمون (*concerned*) بمجموعة من الازواج ويكون ذلك من خلال اعطاء معادلة يتم من خلالها ايجاد قيم y مقابل قيم معطاة لـ x , كما في المثال التالي:

مثال 3-44:

ان $Q=\{(x,y) \mid y=4 - 2x , x \text{ an integer}\}$ تمثل علاقة فيها الازواج المُرتَّبة هـي (x,y) , حيـث ان x تمثل اعداد صحيحة وان المعادلة هي $(y = 4 - 2x)$. ان الزوج $(1,2)$ هو من عنـاصر العلاقـة Q بسـبب انـه اذا كانـت $(x = 1)$ فـان $(y = 4 - 2.1 = 2)$. وبـذلك فـان قيمـة $(y = 2)$ تقابـل قيمـة $(x = 1)$. ان العناصر الاخرى لـ Q هي: $(3,-2),(-1,6),(0,4)$, حيث في كل حالة عوضّنا x بعدد صحيح وحسـبنا قيمـة y من الصيغة $(y = 4 - 2x)$. وبشكل اكثر وضوحاً ان Q لها عناصر مالانهاية .

تعريف 3-13 :

ندعو مجموعة الزوج المُرتَّب (a,b) , حيث a في المجموعة A و b في B , علاقة او علاقة ثنائية Binary (relation) من A الى B , وندعو A بمجال العلاقة (*The **domain** of the relation*) , و B مدى العلاقة (*The **range** of the relation*).

بتعبير اخر ان العلاقة **R** من المجموعة **X** الى المجموعة **Y** هي مجموعة جزئية (*subset*) من حاصل الضرب الكارتيزي (Cartesian product) (XxY). واذا كانت x تنتمي الى **R** $(x \in R)$, فاننا نكتبها (xRy) ونقول ان x مرتبطة بـ y . وفي حالة $(X = Y)$, فاننا ندعو **R** علاقة ثنائية (binary relation) على المجموعة **X** . وان المجموعة $\{ x \in X \mid (x,y) \in R \quad for\ some\ y \in Y \}$ تدعى المجال لـ **R** . وان المجموعة $\{ y \in Y \mid (x,y) \in R \quad for\ some\ x \in X \}$ تدعى المدى لـ **R** .

في المثال(3-44) ان مجموعة المتغير x هي اعداد صحيحة وان المجموعة لكل القيم الممكنة لـ x تسمى مجال العلاقة (**Domain**) . اذن مجال العلاقة **Q** هو مجموعة الاعداد الصحيحة $\{-3,2,4\}$. ونسبة الى قيم x فاننا نجد قيم y . وتسمى كل قيم y الممكنة بمدى العلاقة (**Range**) وهي تمثل كل الاعداد الصحيحة الزوجية $(-2,6,4,2)$. اما مجال ومدى العلاقتين في المثال (3-43) فهما : $\{-3,2,4\}$ و $\{-5,-3,4\}$ على التوالي.

مثال 3-45:

اوجد المجال والمدى للعلاقة التالية:

$$Q = \{(x,y) \mid y = \sqrt{x}$$

ان المتغير x لايمكن تعويضه بعدد سالب لكونه تحت الجذر التربيعي . اذن قيم x هي اي عدد صحيح غير سالب . وبتعويضها نحصل على قيم y . اذن المجال هو مجموعة كل الاعداد الصحيحة الغير سالبة وان المدى هو ايضاً نفس المجموعة .

مثال 3-46:

اوجد المجال والمدى للعلاقة التالية:

$$R = \{(x,y) \mid y = 2x\}$$

ان المتغير x بالامكان تعويضه باي قيمة حقيقية , بسبب ان العدد الحقيقي المضروب في **2** يعطي عدد حقيقي ايضاً . اذن المجال للعلاقة **R** هو :

$$\{ x \mid x\ a\ real\ number\}$$

اما المدى فهو كذلك مجموعة الاعداد الحقيقية , بسبب ان مضاعفة عدد حقيقي **(2x)** هو عدد حقيقي اخر .وعلى اساس ذلك فان المدى هو:

$$\{ y \mid y \text{ a real number}\}.$$

مثال 3-47:

اوجد المجال والمدى للعلاقة التالية:

$$S = \{ (x,y) \mid y = x^2 + 1$$

ان المتغير x بالامكان تعويضه باي قيمة حقيقية, بسبب ان نتيجة تربيع اي عدد حقيقي ثم اضافة **1** اليه يعطيً عدد حقيقي ايضا . اذن مجال العلاقة **S** هو $\{ x \mid x \text{ a real number}\}$. من جانب اخر ان مربع اي عدد حقيقي لايكون سالباً ابداً $(x^2 \geq 0)$. اذن $(x^2 + 1 \geq 1)$.

اذن المعادلة $y = x^2 + 1$ تتضمن (***implies***) $(y \geq 1)$, وبالتالي فان مجال العلاقة S هو $\{y \mid y \geq 1\}$.

مثال 3-48:

اوجد المجال والمدى للعلاقة التالية:

$$R = \{ (x,y) \mid y \leq x, x \text{ and } y \text{ positive integers less than } 5\}$$

حيث x و y هما اعداد صحيحة موجبة .

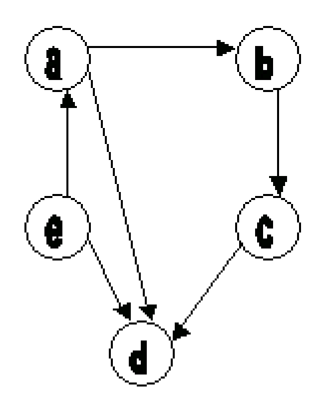
ان المجال كما وُصِف هو مجموعة العناصر x للاعداد الصحيحة الموجبة اقل من **5** اي : $\{1,2,3,4\}$. وعند تعويض قيم x نجد انه عندما $(x = 1)$ فان $(y = 1)$ ايضاً , وعندما $(x = 2)$ فان $(y = 1 \text{ or } 2)$, وعندما $(x = 3)$ فان $(y = 1,2 \text{ or } 3)$. وعندما $(x = 4)$ فان $(y = 1,2,3 \text{ or } 4)$, وبالتالي فان المدى هو نفس مجموعة المجال اي $\{1,2,3,4\}$ (لاحظ اننا لم نكرر عناصر المدى المكررة).

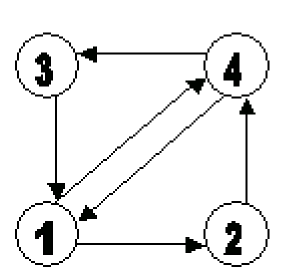
واخيراً يمكننا ادراج عناصر العلاقة **R** من الازواج المُرتَّبة كمايلي:

$$\{(1,1),(2,1),(2,2),(3,1),(3,2),(3,3),(4,1),(4,2),(4,3),(4,4)\}$$

مثال 3-49:

اكتب العلاقة طبقاً الى المخططات المباشرة(Directed Graphs) المبينة بالشكل (3-7) التالية:





ان العلاقات للمخططات اعلاه هي كالاتي:

$\{(1,2), (2,4), (1,4), (4,1), (4,3), (3,1)\}$ $\{(a,b),(b,c),(c,d),(a,d),(e,a),(e,d)\}$

شكل (3-7) المخططات المباشرة(Directed Graphs) للمثال 3-49

مثال 3-50:

اكتب مجموعة الازواج المُرتَّبة للعلاقة بين **x** و **y** المعطاة بالمعادلة التالية:

$$y = \pm x$$

وذلك من المجموعة $\{0 ,1,2,3,4\}$ التي تمثل المجموعة الكلية لـ x الى المجموعة $\{-4,-3,-2,-1,0,1,2,3,4\}$ التي تمثل المجموعة الكلية لـ y .

الحل:

ان العلاقة في هذا المثال تتمثل بالازواج المُرتَّبة التالية:

$$\{(0,0),(1,1),(1,-1),(2,2),(2,-2),(3,3),(4,4),(4,-4)\}$$

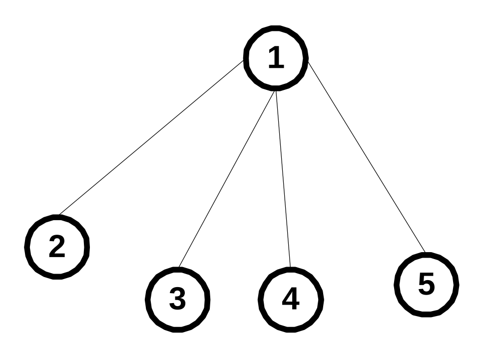
وذلك لان قيمة واحدة لـ x تقابلها قيمتين لـ y حسب المعادلة التي تربط بين x و y, ($y=\pm x$)

مثال 3-51:

ارسم المخطط (graph) على المجموعة $\{1,,2,,3,4,5\}$ لمجموعات الحدود (edge sets) التالية:

$$a\text{-}\{\{1,2\},\{1,3\},\{1,4\},\{1,5\}\}.$$
$$(b)\text{-}\{\{1\},\{1,2\},\{2,3\},\{3\},\{4\},\{4,5\},\{5\}\}.$$

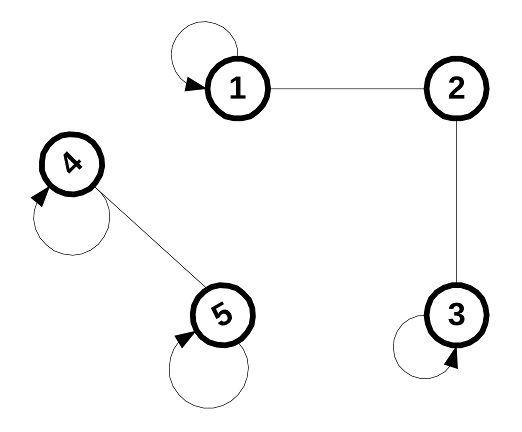
ان مخطط العلاقة في a يمثل مخطط غير مباشر (*undirected graph*) , كما في الشكل (3-8) كما يلي :



$$a\text{-}\{\{1,2\},\{1,3\},\{1,4\},\{1,5\}\}.$$

شكل(3-8) يبين مخطط العلاقة a في المثال 3-51

ومخطط العلاقة في b هوكما في الشكل (3-9) :



(b)-{{1},{1,2},{2,3},{3},{4},{4,5},{5}}.

شكل(3-9) يبين مخطط العلاقة b في المثال 3-51

مثال 3-52:

افرض ان $X = \{2,3,4\}$ وان $Y = \{3,4,5,6,7\}$ فاذا عرّفنا علاقة **R** من X الى Y بالعلاقة التالية:

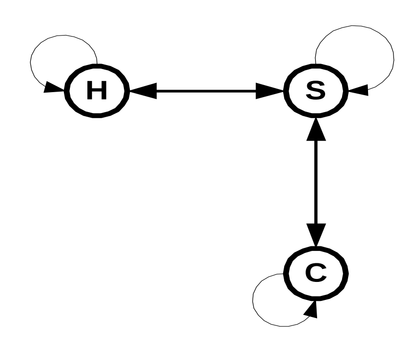
$$(x,y) \in R \quad If \ \ x \ divides \ y \ (with \ zero \ remainder)$$

فاننا نحصل على العلاقة التالية بالازواج المُرتّبة : $R = \{(2,4),(2,6),(3,3),(3,6),(4,4)$

اذن مجال الدالة هو المجموعة $\{2,3,4\}$ ومداها هو المجموعة $\{3,4,6\}$

3-12-3 العلاقة المتماثلة Symmetric relation

يقال لعلاقة **R** بانها متماثلة اذا عندما يكون الزوج (x,y) في **R** اي $(x,y) \in R$, فان الزوج (y,x) ايضا في **R** اي $(y,x) \in R$. ومعنى اخر ان علاقة ما هي متماثلة اذا حينما x ترتبط بـ y بالعلاقة **R** ونرمز لها بـ $R(xRy)$, فان y هي ايضاً ترتبط بـ x بالعلاقة $R(yRx)$. ولتمثيل العلاقة المتماثلة فان كل سهم يقترن بسم اخر وبالاتجاه المعاكس انظر الشكل(3-10) التالية:



الشكل(3-10) يبين العلاقة المتماثلة

تعريف 3-14:

ان العلاقة **R** هي علاقة تكافؤ (*Equivalence relation*) اذا امتلكت الخصائص الثلاثة التالية:

1- خاصية الانعكاس (*reflexive property*) : اذا كان **a** هو عنصر ينتمي الى مجال العلاقة **R** , فان الزوج (a,a) هو عنصر العلاقة **R** اي ينتمي لها. اي ان العنصر **a** يرتبط مع نفسه .ونرمز لتلك العلاقة بـ

$$R \quad is\ reflexive \quad if\ (x,x)\,R\ \ for\ each\ x\ in\ R\ .$$

2- خاصية التماثل (*Symmetric property*), اذا كان الزوج (a,b) هو عنصر من عناصر العلاقة **R** , فان الزوج (b,a) هو ايضاً من عناصر العلاقة **R** .

3- الخاصية الانتقالية (*Transitive property*): اذا كان (a,b) و (b,c) كلاهما من عناصر العلاقة **R** فان (a,c) هو من عناصر **R** ايضاً.

مثال 3-53:

افترض العلاقة التالية :

$$R = \{(x,y) \mid y = x, x \quad is\ an \quad integer\}$$

ولبيان ان **R** هي علاقة تكافئ فانه يجب فحص الخصائص الثلاثة السالفة الذكر وكمايلي:

1- فحص خاصية الانعكاس (*Reflexive*): اذا كان **a** هو عدد صحيح وهو يمثل x هنا حسب العلاقة , فان $(a = a)$, لذا فان (a,a) هو عنصر من عناصر **R** المبينة في العلاقة اعلاه.اي ان العنصر a يرتبط مع نفسه.

2- فحص خاصية التماثل (*Symmetric*) :افترض ان الزوج (a,b) ينتمي الى **R** وهذا يعني $(x=a)$,و $(y=b)$, حيث ان الزوج (a,b) يناظر الزوج (x,y). فطبقاً الى تعريف **R** اعلاه, فان **a** و **b** هما عددان صحيحان وذلك عندما **x** هو عدد صحيح وعندما يساوي **y** ,فان **y** تصبح عدد صحيح ايضاً, اي ان $(b=a)$. لكن اذا كان $(b=a)$ فان $(a=b)$, لذلك فان (b,a) هو ايضاً ينتمي الى **R**.

3- فحص خاصية الانتقالية (*Transitive*) : افترض ان (a,b) و (b,c) كلاهما ينتمي الى العلاقة **R** , فان $(b=a)$ وان $(c=b)$. لذا فان $(c=a)$.اذن (a,c) ينتمي ايضاً الى **R**.

وبما ان **R** اعلاه تمتلك الخواص الثلاثةاعلاه,اذن هي علاقة تكافئ (*Equivalence relation*).

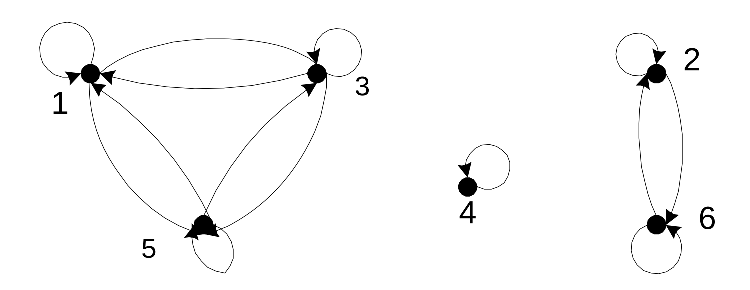
▲

من جانب اخر , يمكننا من خلال مخطط العلاقة المباشر (الموجّه) (***directed graph***) , ان نحدد ان علاقة ما **R** هي علاقة تكافئ وذلك كما يلي:

اذا اعتبرنا ان المجموعة $S=\{\{1,3,5\},\{2,6\},\{4\}\}$ هي جزء (partition) من المجموعة $X=\{1,2,3,4,5\}$, فان العلاقة **R** هي علاقة تكافئ على **X** بسبب انه اذا كانت **S** جزء من مجموعة **X** , فان تعريف العلاقة xRy لتعني انه لبعض المجموعة **S** في **S** ان كلا من x و y ينتمي الى **S** , ومن ثم فان **R** هي علاقة تكافئ. اي ان العلاقة **R** تحوي الازواج المُرتَّبة $(1,1),(1,3),(1,5)$ بسبب ان $\{1,3,5\}$ هي في المجموعة S. ان العلاقة الكاملة **R** هي:

$$R=\{(1,1),(1,3),(1,5),(3,1),(3,3),(3,5),(5,1),(5,3),\\ (5,5),(2,2),(2,6),(6,2),(6,6),(4,4)\}$$

واذا رسمنا العلاقة **R** كما في الشكل التالي , فاننا نجد في كل نقطة (*Vertex*) يوجد حلقة (*loop*) اي وجود الازواج $(1,1),(2,2),(3,3),(4,4),(5,5),(6,6)$ وهذا يعني انها علاقة انعكاس (Reflexive) , وانها علاقة تماثل (Symmetric) لانه لكل حد متجه (*directed edge*) من v الى w انه يوجد ايضاً حد متجه من w الى v . وانها علاقة انتقال (Transitive) بسبب انه اذا وجد حد متجه من **x** الى **y** , ومن **y** الى **z** فانه يوجد حد متجه من x الى **z** . انظر الشكل (3-11) .



الشكل (3-11) يبين مخطط العلاقة المباشر (المتجه) (*directed graph*) لـ **R** اعلاه

مثال 3-54:

مَن مِن العلاقات التالية يملك خاصية الانعكاس على المجموعة $\{1,2,3\}$ ؟

$$a - \{(1,1),(1,2),(2,2),(1,3),(3,2),(3,3)\}$$
$$b - \{(1,1),(2,2),(2,3),(3,2),(3,1),(1,3)\}.$$

ان العلاقة **a** تمثل ذلك, لان الازواج التالية تنتمي اليها .(1, 1) , (2, 2) , and (3, 3) . اذن هي صحيحة وبذلك فان : $(x,x) \in R$, بينما العلاقة **b** لاتمثل ذلك لعدم احتواءها على الزوج (3,3).

مثال 3-55:

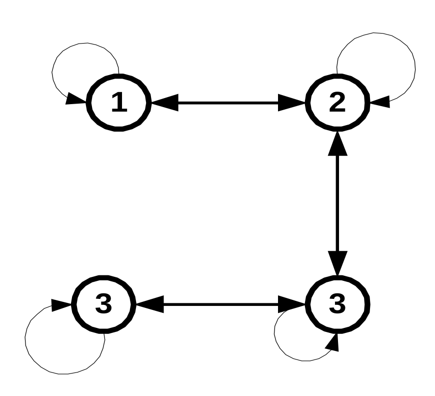
اكتب الازواج المرتبة للعلاقة التي تصف العلاقة بين x و y والمعطاة بالعلاقة التالية:

ان **x** ترتبط بـ **y** اذا كانت القيمة المطلقة للفرق بين x و y اصغر او تساوي 1 ($|x - y| \leq 1$) على المجموعة $\{1,2,3,4\}$. ثم ارسم المخطط المباشر (Digraph) لتلك العلاقة.

الحل:

عندما $(x = 1)$ فان $(y = 1,2)$, وعندما $(x = 2)$, فان $(y = 1,2,3)$, وهكذا الخ..............وبالتالي فان العلاقة هي:

$R = \{(1,1),(1,2),(2,1),(2,2),(2,3),(3,2),(3,3),(3,4),(4,3),(4,4)\}$ ومخطط العلاقة هوكما في الشكل (3-12) :



الشكل (3-12) يبين المخطط المباشر (Digraph) للعلاقة في المثال 3-55

من جانب اخر اذا كانت العلاقة **R** تمتلك الخاصية الانتقالية وخاصية الانعكاس لكنها غير متماثلة (*antisymmetric relation*) فاننا نطلق عليها علاقة الترتيب الجزئي (*partial orders*) اي تستخدم لترتيب عناصر المجموعة , على سبيل المثال, اذا عرّفنا العلاقة **R** على مجموعة الاعداد الصحيحة او الاعداد الصحيحة الموجبة بالعلاقة :

$$(x,y) \in R \quad if \quad x \leq y, x \quad or \quad If \ x \ \ divides \ y(evenly) \quad \}$$

فان **R** تدعى علاقة الترتيب الجزئي (*Partial Order*) .

3-13 الدالة Function :

في عدة مجالات في الرياضيات وتطبيقاتها , ان نوع العلاقة الاكثر اهمية هي العلاقة التي فيها كل قيمة من المجال تقابلها قيمة واحدة من المدي لها , ويسمى مثل تلك النوع الخاص من العلاقة بالدالة (*function*) , ومثال على ذلك العلاقة : $R = \{(x,y) \mid y = 2x\}$, حيث من الواضح ان قيمة واحدة لـ x تقابلها قيمة واحدة لـ *y* بسبب ان المعادلة التي تربطهما هي : $y = 2x$.

لكن العلاقة المبينة في المثال (3-48) التالية:

$$R = \{(x,y) \mid y \leq x, x \ and \ y \ \ positive \ integers \ less \ than \ 5\}$$

ليست دالة بسبب ان

لكل قيم x : (2, 3, و 4) توجد اكثر من قيمة لـ y , وذلك لأن $y \leq x$.

وبما ان الدالة هي علاقة فانه يمكننا رسمها بيانياً باستخدام نظام الاحداثيات الكارتيزي مستخدمين المحور السيني (*x-axis*) والمحمور الصادي (*y-axis*). وبسبب ان الدالة تمتلك قيمة واحدة لـكل قيمة لـ x في المجال , فان الرسم البياني لايمكن قطعه في اكثر من نقطة واحدة بواسطة اي خط عمودي (*vertical line*).

1-13-3 رموز الدالة : Function Notations

يمكننا الرمز للدالة كما هو في الصيغ التالية:

$$f(x) = 2 - 3x, \qquad f(x) = 2 - x2, \qquad G(x) = 2x, \ or \qquad G = \{(x, y) | y = 2x\}$$

وفي الجبر والرياضيات ان الدالة هي علاقة (*relationship*) تعطى بصيغة مثل :

$$f(x) = x + 2 \qquad or \quad y = x + 2$$

كما ان الدالة يمكن أن تكون لمتغير واحد آو اثنان أو ثلاث كمايلي:

ان الدالة $f(x, y) = x\,y + x^2$ هي دالة لمتغيرين وان مجالها هو مجموعة كل الازواج المرتبة (x, y) مـن الاعداد. وان الدالة $g(x,y,z) = x + 2y + 3z$ هي دالة لثلاثة متغيرات x , y و z وان مجالها هـو مجموعـة كل الثلاثيات المرتبة ordered triples من الاعداد .

من جانب اخر , يمكننا ان نبين انه صيغتين مختلفتـين يصـفان نفس العلاقـة بـين المجموعـة $\{-1,0,1,2\}$ والمجموعـة $\{1,2,3,4\}$. , مثال على ذلك الصيغتين التاليتين:

$$y = x + 2, \qquad y = x^4 - 2x^3 - x^2 + 3x + 2$$

وبما ان الصيغتين مختلفيتن , اذن لابد من البحث عن تعريفات للدالة غير تعريفاتها بطريقة الصيغ الرياضـية اعـلاه وبـدلاً من ذلك نستخدم مبدأ العلاقة (*relation*) لتعريف الدالة كمايلي:

1- ان دالة من المجموعة **D** الى المجموعة **R** هي علاقة من **D** الى **R** , مثل هذا ان كل x في **D** ترتبط بقيمة واحدة فقط وفقط واحدة في **R** . ان **D** يدعى المجال (*Domain*) و **R** تدعى المدى (*Range*) للدالة .

2- تُعرّف العلاقة (*Relation*) بانها مجموعة من الازواج المُرتَّبة (*Ordered Pairs*), لذا ستُعرَّف الدالة بانها حالـة خاصـة من مجموعة من الازواج المُرتَّبة .

ان الرمز الاساسي للدالة هو (f) , ولو عرَّفنا دالة من **D** الى **R** فيمكن كتابتها رمزياً بالشكل التالي:

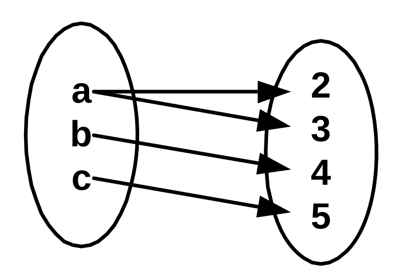
$$f : D \rightarrow R$$

ان الطريقة القياسية لاظهار الدالة بشكل مرئي هو استخدام مخطط السحابة (*Cloud diagram*) كـما في الشـكل ادنـاه , حيث ان كل سهم يمثل زوج مُرتَّب واحد في العلاقة (*relation*) ويرمز له بـ $(x, f(x))$.

مثال 3-56:

بـــيّن ان العلاقـــة $\{(a,2),(a,3),(b,4),(c,5)\}$ هـــي ليســـت دالـــة مـــن $D=\{a,b,c\}$ الى $R=\{2,3,4,5\}$

ان العددين **2** و **3** في المجال يرتبطان بـالنقطة **a** في المجال (اي سهمين يخرجان من **a**) وهذا عكـس تعريـف الدالـة . اذن العلاقة هنا ليست دالة . انظر الشكل (3-13) .

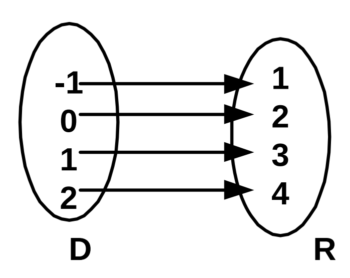


الشكل (3-13) يمثل العلاقة في المثال 3-56

مثال 3-57:

اذا كانت $f(x)= x+2$ تستخدم كصيغة (*formula*) تصف الدالة f من المجموعة $\{-1,0,1,2\}$ الى المجموعة $\{1,2,3,4\}$. ماهي العلاقة التي تُعرّف f ؟

ان العلاقة هي: $f=\{(-1,1),(0,2),(1,3),(2,4)\}$ وتُوضّح بالشكل (3-14) التالي:



الشكل (3-14) يمثل العلاقة في المثال 3-57

تعريف 3-15:

ان الدالة هي نوع خاص من العلاقة فيها قيمة واحدة من المجال تقابلها قيمة واحدة من المدى.

A function is a relation such that to each domain value there corresponds exactly one range value.

من جانب اخر, ان الدالة (*function*) التي هي حالة خاصة من العلاقة الرياضية(*relation*) تعطي لكل عنصر من المجموعة **X** عنصر واحد بالضبط من المجموعة **Y** . ان الدوال تستخدم بشكل واسع في الرياضيات المتقطعة , على سبيل المثال , الدوال تستخدم لتحليل الوقت الضروري لتنفيذ الخوارزميا ت (*Algorithms*)

ومن الامثلة الكثيرة على الدالة التي نتعامل معها يومياً هي: مثلاً عدد الاميال المقطوعة لكل غالون وقود هو دالة للسرعة التي نقود بها السيارة . ومقدار ماندفعه لشراء الحاجيات اليومية من الاسواق هو دالة لوزنها , وكذلك ثمن ارسال رسالة او طردا بريدياً في البريد هو دالة لوزنهما....وهكذا يمكنك تصور امثلة عديدة حول الدالة.

مثال 3-58:

هل العلاقة $\{(x,y) \mid y = 2x+3, x \text{ a real number}\}$ هي دالة ؟, وضِّح ذلك .

اذا كانت x عدد حقيقي , فان y تمثل العدد الحقيقي $2x+3$, فمثلاً لو كانت $x = 2$ فان $y = 7$, واذا كانت $(x = -1/2)$ فان $y = 2$. وبشكل واضح ان قيمة واحدة لـ x , ان التعبير الرياضي $2x+3$ يعطينا قيمة واحدة وفقط واحدة لـ y . وعلى اساس ذلك ان العلاقة المعطاة يكون لها قيمة واحدة للمدى مقابل قيمة واحدة للمجال. اذن العلاقة تمثل دالة.

مثال 3-59:

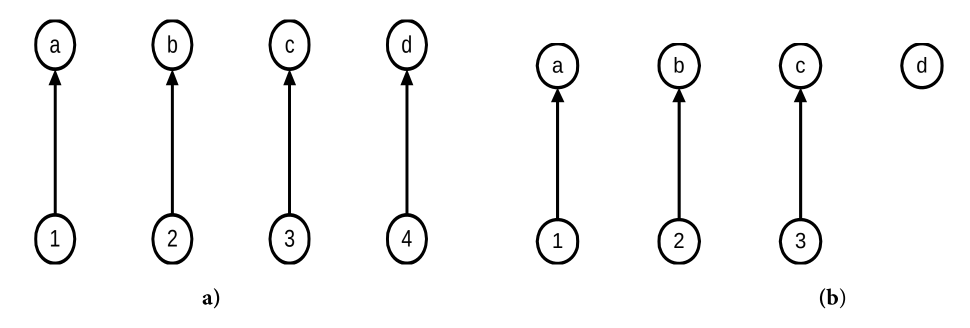
هل العلاقة $\{(x,y) \mid x = y^2 + 1\}$ هي دالة ؟, وضِّح ذلك .

ان مجال العلاقة هذه هو المجموعة $\{(x \mid x \geq 1\}$, ولو اخذنا $x = 5$ مثلاً , فان المعادلة $x = y^2 + 1$ تعطينا $5 = y^2 + 1$.

اذن , $y^2 = 4$, اي ان قيم y هي $y = \pm 2$.اذن الزوج ($(5,-2)$ والزوج $(5,2)$ كلاهما للعلاقة اعلاه. وعلى اساس ان قيمتين من المدى للعلاقة هناك قيمة واحدة للمجال, وهذا يخالف تعريف الدالة . اذن العلاقة $\{(x,y) \mid x = y^2 + 1\}$ لاتمثل دالة.

2-13-3 خصائص الدالة Function properties

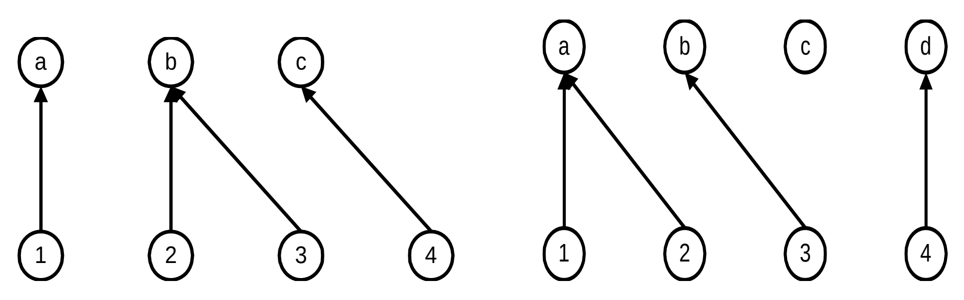
1 - ان الدالة f تدعى (*injection*) او دالة واحد الى واحد (*one-to-one function*) اذا كانت عناصر مختلفة من المجال ترتبط بعناصر مختلفة من المدى (*range*) . هكذا ان دالة واحد الى واحد اذا كان كل عنصر في مدى الدالة يرتبط على ابعد تقدير عنصر واحد من المجال (*domain*) . لذا فان الدالة في الشكل (3-15) (**a**) و (**b**) هما دالّتا واحد الى واحد انظر الشكل (3-15) .



الشكل (3-15) يمثل خصائص الدالة

ان المجالين للدالتين في (a) و (b) اعلاه , هما $\{1,2,3,4\}$ و $\{1,2,3\}$ على التوالي.

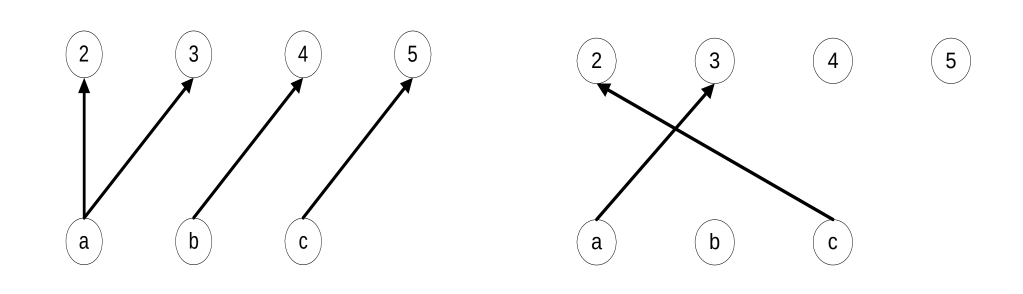
2- ان الدالة f تدعى (*surjection*) او (*onto function*) اذا كان كل عنصر في المدى يرتبط على الاقل بعنصر واحد من المجال. وهنا يعني ان من الممكن ان نقطتين من المجال تحققان نقطة واحدة من المدى للدالة. انظر الشكل(3-16) ادناه:



الشكل (3-16) يمثل خصائص الدالة الاخرى

مثال 3-60:

ان العلاقات في المخططات المتجة التالية لاتمثل دوال بسبب انه سهمين تخرج من النقطة **a** ولايوجد سهم يخرج من **b** . انظر الشكل (3-17)



Two arrows leave a ***no arrows leave b***

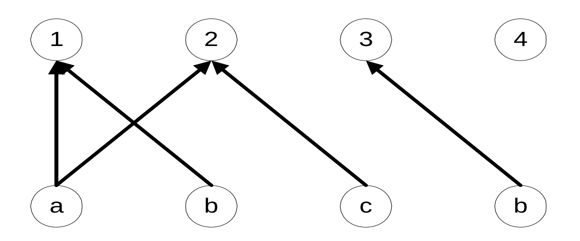
الشكل (3-17) يبين العلاقات في المثال 3-60

مثال 3-61:

كما سبق فانه بالنسبة الى الدوال فانها اما دالة واحد الى واحـد *(one-to one)* , او دالـة *(onto) function* , او كلاهـما *(one-to-one or onto function)* وتدعى *(bijection function)* . اذن اي من العلاقات المعطاة ادنـاه تمثـل دالـة مـن المجال **D** المعطى الى المدى **R** المعطى ايضاً.

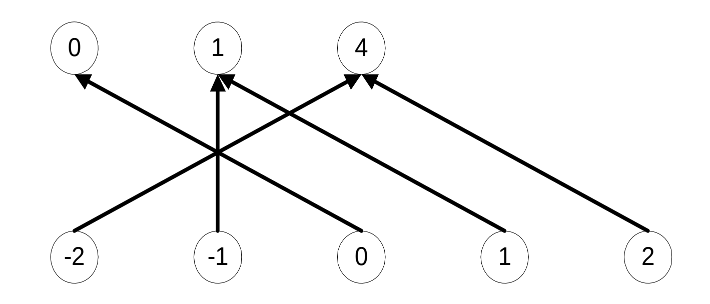
$$(a)\text{-}\ \ f=\{(a,1),(a,2),(b,1),(c,2),(d,3)\}. \qquad D=\{a,b,c,d\} \qquad R=\{1,2,3,4\}$$

ان العلاقة **(a)** لاتمثل دالة لان النقطتين 1 و 2 كلاهما يرتبط بـالنقطة **a** (انظر الشكل(3-18)



الشكل(3-18) يمثل العلاقة **a** في المثال 3-61.

$$(b)\text{-}\ f(x)=x^2 \qquad D=\{-2,-1,0,1,2\} \qquad R=\{0,1,4\}$$



الشكل(3-19) يمثل العلاقة **b** في المثال 3-61.

ان العلاقة **(b)** تمثل دالة لأن لكل x , توجد قيمة واحدة $f(x)$, كما انها دالة (*onto function*) بسبب ان كل من **0**, **1**, و **4** هي نتيجة تربيع قيمة المجال , اي هناك قيمتان لـ x يقابلها قيمة واحدة لـ y. وان f هي ليست دالة واحد الى واحد(*one-to one*) بسبب ان $f(-2)=4$, و $f(2)=4$.

14-3 تركيب وعكس الدوال: Composition and inverse of functions

افترض ان R_1 هـي علاقة مـن X الى Y وان R_2 هـي علاقة مـن Y الى Z , فـان تركيـب R_1 و R_2 يرمـز لـه بــ $R_2 \circ R_1$ وهو علاقة من X الى Z ويرمز لها بـ

$$R_2 \circ R_1 = \{(x,y) \mid (x,y) \in R_1 \quad and \quad (x,y) \in R_2 \; for \quad some \quad y \in Y$$

وباستعمال الدوال فاننا نجمعها بطرق مختلفة , على سبيل المثال, اذا كانت:

$$f(x) = \sqrt{x} \quad \text{و} \quad g(x) = 1 - x^2$$

فان :

$$f(g(x) = \sqrt{1-x^2} = f \circ g(x)$$

حيث :

$$f(x) = \sqrt{x} \therefore f(g(x)) = \sqrt{g(x)} = \sqrt{1-x^2}$$

وهذه تعطي ازواجاً من (x, y) اذا كان الطرف الايمن يساوي y . ان تركيب الدالة f مع الدالة g هو علاقة (relation جديدة يرمز لها بـ $f \circ g(x)$ وتقرأ ("**f circle g**"), أو (***f composed with g***) التي تحتوي الازواج (x, y) اذا وفقط اذا $y = f\,(g(x))$ التي تقرأ $(f \; of \; g \; of \; x)$.

مثال 3-62:

لكل زوج من الدوال المعطاة ادناه اوجد $f \circ g$.

(a)- $f(x) = 2x + 3; \quad g(x) = \frac{1}{2}x$

(b)- $f = \{(1,2),(2,3),(3,4),(4,6)\}, \qquad g = \{(a,1),(b,2),(c,4),(d,3)\}$

الحل:

$$a: f \circ g(x) = 2(\frac{1}{2}x) + 3 = x + 3$$
$$b: f \circ g(x) = \{(a,2),(b,3),(c,6),(d,4)\}$$

مثال 3-63:

في زوج الـدوال ادنـاه , ان صـورة الدالـة g (image) هـي مجموعـة جزئيـة مـن مجـال الدالـة f , وهـذا لـكي تُعـرِّف $f \circ g(x)$. لذا اعطي افضل وصف باستطاعتك لدالة التركيب (*composition function*) : $f \circ g(x)$

$$a) - f = \{(1,-1)(2,-2)(3,-3)(4,-4)(5,-6)\} \qquad g = \{(1,3),(2,4),(3,5)\}$$

ان مجال الدالة f هو المجموعة $\{1,2,3,4,5\}$, ومجال الدالة g هو $\{1,2,3\}$, اما صورة الدالة g فهي تمثل المدى فيها اي عناصر y في كل زوج مرتب فيها وهي $\{3,4,5\}$, ونلاحظ ان مجموعة الصورة هي مجموعة جزئية من مجموعة مجال f. اذن يمكننا تعريف $f \circ g(x)$.

والان كيف نجد $f(g(x))$ لكل x في في المجال لـ g.؟

ويتم ذلك من خلال :

1- ايجاد $g(x)$

2- تعويض $g(x)$ في الصيغة للدالة f

ولوصف العلاقة $f \circ g(x)$ كمجموعة من الازواج المرتبة فاننا نحتاج ان نعرف $f \circ g(1)$, $f \circ g(2)$ و $f \circ g(3)$ لان لدينا مجال لـ g هو $\{1,2,3\}$ وهو يمثل عناصر x. ونجد كمايلي:

$$f(g(1)) = f(3) = -3 \qquad this\ give\ us\ the\ pair \quad (1,-3)$$
$$f(g(2)) = f(4) = -4 \qquad this\ give\ us\ the\ pair \quad (2,-4)$$
$$f(g(3)) = f(5) = -6 \qquad this\ give\ us\ the\ pair \quad (3,-6)$$

لاحظ ان $f(g(1)) = f(3) = -3$ قد اعطتنا (-3) , وعند التفتيش عن -3 في الدالة ***f*** نجد انها في الزوج $(3,-3)$ اي ان مع (-3) العدد 3 الذي هو لـ **x** , وبالتالي نبحث عن العدد **3** في الدالة ***g*** فاننا نجده في الزوج $(1,3)$, اذن ناخذ منه العدد **1** الذي يمثل **x** لنحصل اخيراً على الزوج $(1,-3)$ الذي هو للعلاقة $f \circ g(x)$, وبنفس الطريقة نجد بقية الازواج .

اذن مجموعة الازواج المُرتَّبة للعلاقة $f \circ g(x)$ هي: $f \circ g(x) = \{(1,-3),(2,-4),(3,-6)\}$

▲

مثال 3-64:

مرة ثانية , في كل زوج من الدوال ادناه , ان صورة الدالة ***g*** (*image*) هي مجموعة جزئية من مجال الدالة ***f*** , لكي تُعرِّف $f \circ g(x)$. لذا اعطي افضل وصف باستطاعتك لدالة التركيب (*composition function*) : $f \circ g(x)$.

$$1-\ f(y) = y^3 + 1 \ \ and \ \ g(x) = \sqrt[3]{x}$$

افرض ان $y = g(x) = \sqrt[3]{x}$,

اذن :

$$f(g(x)) = f(y) = f(\sqrt[3]{x}) = (\sqrt[3]{x})^3 + 1 = x + 1.$$

▲

اما بالنسبة لزوج الدوال التالي:

$$2-\ f(x) = x^3 + 1, \qquad g(x) = \sqrt[3]{x-1}$$

▲ $$f \circ g(x) = f(g(x)) = f(\sqrt[3]{x-1}) = (\sqrt[3]{x-1})^3 + 1 = x - 1 + 1 = x.$$

مثال: 3-65

اذا كانت R_1 تمثل العلاقة :

$$R_1 = \{(1,2),(1,6),(2,4),(3,4),(3,6),(3,8)\}$$

وكانت R_2 تمثل العلاقة :

$$R_2 = \{(2,u),(4,s),(4,t),(6,t),(8,u)\}$$

فان $R_2 \circ R_1$:

$$R_2 \circ R_1 = \{(1,u),1,t),(2,s),(2,t),(3,s),(3,t),(3,u)\}$$

ولتوضيح ذلك : ان مجال العلاقة R_2 هو المجموعة $\{2,4,6,8\}$, والصورة لها (عناصر y) هي المجموعة $\{u,s,t\}$. واخيراً فان مجال العلاقة R_1 هو المجموعة $\{1,2,3\}$. اذن لايجاد العلاقة $R_2 \circ R_1$, فاننا نحتاج لايجاد ازواج $R_2 \circ R_1(1)$, $R_2 \circ R_1(2)$, و $R_2 \circ R_1(3)$ حسب مجال R_1 .

لاحظ ان $R_2(R_1(1)) = R_1(1) = 2,6$ قد اعطتنا $(2,6)$,اي مع الـ 1 في R_1 يوجد الزوجين $(1,2),(1,6)$ وعند التفتيش عن $2,6$ في الدالة R_2 نجد انهما في الازواج $(2,u),(6,t)$ اي ان مع (u,t) ناخذ العدد 1 من R_1 الذي هو لـ x , لنحصل اخيراً على الازواج $(1,u),1,t)$ الذي هو للعلاقة $R_2 \circ R_1$, وبنفس الطريقة نجد بقية الازواج .وتكون العلاقة النهائية :

$$R_2 \circ R_1 = \{(1,u),1,t),(2,s),(2,t),(3,s),(3,t),(3,u)\}$$

3-14-1 الدوال العكسية Inverse Functions

عندما تكون f و g دالتين , و كانت $f(g(x)) = x$, و $g(f(y)) = y$ لكل x في مجال g , ولكل y في مجال f فاننا نقول ان f و g عكس بضعهما الاخر (يعني f عكس g و g عكس f) . ونرمز لتلك العلاقة بـ $f = g^{-1}$ and $g = f^{-1}$ ونقرأ f^{-1} عكس وليست مرفوعة الى الأُس (-1) .

فلو كانت f علاقة من X الى Y , فان عكس f (***inverse of f***) يرمز له بـ f^{-1} هو العلاقة من Y الى X. ونعبر عنها كمايلي:

$$f^{-1} = \{(y,x) | (x,y) \in f\}$$

مثال 3-66:

لكل زوج من الدوال التالية , حدِّد فيما لو كانت الدالة f عكس الدالة g .

(a)- $f(x) = x^3 + 1;$ $\qquad g(x) = \sqrt[3]{x-1}$

الحل:

افرض ان $y = f(x)$, اذن $y = x^3 + 1$, اذن قيمة x ستكون $x = \sqrt[3]{y-1}$, وعليه فان

$$f^{-1}(x) = \sqrt[3]{x-1} = g(x)$$

واخيراً نستنتج ان الدالة f عكس الدالة g .

مثال 3-67:

اذا كانت $f(x) = \sqrt[3]{x} - 1$ و $g(x) = (x-2)^3 + 1$, فاوجد $f^{-1}(x)$ و $g^{-1}(x)$.

الحل:

1- افرض ان $y = g(x)$, اذن $y = (x-2)^3 + 1$, اذن قيمة x ستكون $x = \sqrt[3]{y-1} + 2;$, وعليه فان

$$f^{-1}(x) = \sqrt[3]{x-1} + 2 = g^{-1}(x)$$

2- افرض ان $y = f(x)$, اذن $y = \sqrt[3]{x} - 1$,

وان: $x = (y+1)^3 = f(y);$

وعليه فان :

$$f^{-1}(x) = (x+1)^3 \qquad g^{-1}(x) = \sqrt[3]{x-1} + 2$$

مثال 3-68:

اذا كانت : $R=\{(2,4),(2,6),(3,3),(3,6),(4,4)\}$

فان : $R^{-1}=\{(4,2),(6,2),(3,3),(6,3),(4,4)\}$

3-14-2 الرسم البياني للدالة العكسية The graph of inverse function

اذا كانت $g(x)=x^3$, فان $g^{-1}(y)=\sqrt[3]{y}$, وبالتالي فان العلاقة g تحوي كل الازواج (x,x^3) التي تقابل الازواج (x,y), وان العلاقة g^{-1} تحوي كل الازواج بالصيغة (y,x) وهي $(y,\sqrt[3]{y})=(x^3,x)$.

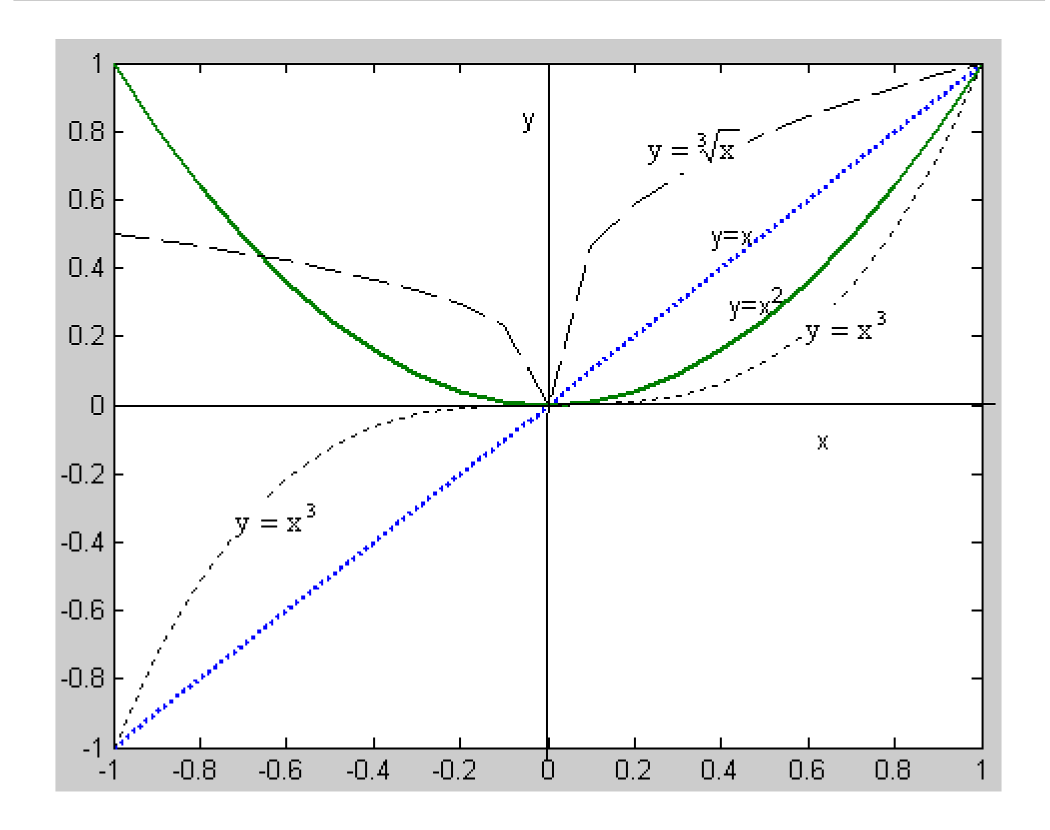
ان العلاقة البيانية ستكون بين النقاط (x,x^3) والنقاط (x^3,x) .

مثال 3-69:

ارسم العلاقتين المعطاة ادناه على نفس المحاور *(axes)* , وبين كيف ان العلاقتين تتعلق بالرسم البياني لـ $y=x$.

$$g(x)=x^3 \quad \text{و} \quad f(x)=\sqrt[3]{x}\ ;$$

ان المخطط البياني ,الشكل (3-20) , ادناه يبين ان الدالتين عكس احدهما الاخرى ,وانهما متماثلتين حول الخط المستقيم $y=x$, حيث ان الدالة f تقع على يمين معادلة الخط المستقيم $y=x$, اما الدالة g فتقع على الشمال منه, ويتم الرسم البياني من خلال ايجاد النقاط (x,y) اي نفرض قيم x ونجد مايقابلها لـ y . اما ايجاد العكس فاننا نعكس قيم الازواج (قيمة x لـ y وقيمة y لـ x) كمايلي:



الشكل (3-20) يبين الرسم البياني للعلاقات في المثال 3-69

3-15 مصفوفات العلاقات Matrices Of Relations

تعتبر المصفوفة الطريقة الملائمة لتمثيل علاقة R مـن X الى Y . مثـل هـذا التمثيـل يسـتخدم بواسـطة الحاسـوب لتحليـل العلاقة . وعلى صفوف المصفوفة (rows) نضع عنـاصر X. ونضـع عنـاصر Y عـلى اعمـدة المصـفوفة وكـلا العنـاصر لـيس بالضرورة ترتيبها بل تُرتَّب بشكل عشوائي. اما مدخلات المصفوفة في الصف x والعمود y فتكون واحد اذا كانت x تـرتبط بـ y (xRy) وصفر عكس ذلك.

مثال 3-70:

لو كانت لدينا العلاقة التالية:

$$R = \{(1,b),(1,d),(2,c),(3,c),(3,b),(4,a)\}$$

ان هذه العلاقة ستكون بالطبع من $X=\{1,2,3,4\}$ الى $Y=\{a,b,c,d\}$ طبقاص الى الترتيب $4,3,2,1$ و c,d,b,a , والمصفوفة التي تمثلها هي كما هو ادناه:

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array}\begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

اما لو اخذنا الترتيب $2,3,4,1$ و c,a,b,d ستكون المصفوفة كالاتي:

$$\begin{array}{c} \\ 2 \\ 3 \\ 4 \\ 1 \end{array}\begin{array}{c} \begin{array}{cccc} d & b & a & c \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix} \end{array}$$

اما مصفوفة العلاقة

$$R=\{(a,a),(b,b),(c,c),(d,d),(b,c),(c,b)\}$$

على المجموعة $Y=\{a,b,c,d\}$ والترتيب c,d,b,a فهي كالاتي:

$$\begin{array}{c} \\ a \\ b \\ c \\ d \end{array}\begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

مثال 3-71:

اما لو كانت لدينا العلاقة R من $X = \{2,3,4\}$ الى $Y = \{5,6,7,8\}$ نسبة الى الترتيب $2,3,4$ و $5,6,7,8$

ومعرَّفة بواسطة xRy اذا كانت x تقسِم y , فان المصفوفة التي تمثلها هي:

$$\begin{array}{c} \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 5 & 6 & 7 & 8 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

تمارين الفصل الثالث

1- Write the following sets using the listing method and using set-builder notation:

a) The set of counting numbers greater than **0** and less than **1**.

b) The set of even counting numbers .

2-If U ={a, b, c, d, e, f} , A = {a, b, c, d}, and B = {b, d, e}. Find the subsets of B and find the following:

$a - \{x \mid x \in U \wedge x \notin B\}$ $\qquad$ $b - \{x \mid x \in A \wedge x \in A\}$

$c - \{x \mid x \in A \wedge x \notin B\}$

3-If U ={a, b, c, d, e, f} , A = {a, b, c, d}, and B = {b, d, e}. Find the subsets of B and find the following:

$a - \{x \mid x \in A \wedge x \in B\}$

$b - \{x \mid x \in U \wedge x \notin A\}$

$c - \{x \mid x \in A \wedge x \notin B\}$

4- Draw Venn diagram to illustrate each of the following sets:

a- $\{x \mid x \in A \;\; or \;\; x \in B\}$, $\{x \mid x \in A \;\; and \;\; x \in B\}$

5- If $A = \{a, b, c\}$ **find the members of the power set(** $p(A)$ **)**

6- Write down the implies relation on the set $S = \{p, q, p \wedge q, p \vee q\}$ **of symbolic statements.**

7- Prove that $\sum_{k=1}^{s}(bk + c) = \frac{1}{2}bs(s+1) + sc$

8- **Write down the ordered pairs of the relation specified:**

$$(r^3 - 8r)^2 = (k^3 - 8k)^2 \text{ on the set } \{-2,-1,0,1,2\}$$

9- **Find the domain and the range of the following relation:**

$$Q = \{(x,y) \mid y = \sqrt{x}\}$$

$$S = \{(x,y) \mid y = x^2 + 1\}$$

10- **Decide Whether the given relation is a function. State the reason for your answer in each case.**

1- $Q = \{(x,y) \mid x = y^2\}$ 2- $R = \{(x,y) \mid y = x^2\}$

11- **Write the equivalent classes of the equivalent relation given :**

$(x,y) \in R$ if $x^2 = y^2$ on the set {-2, -1, 0, 1, 2, 3 }

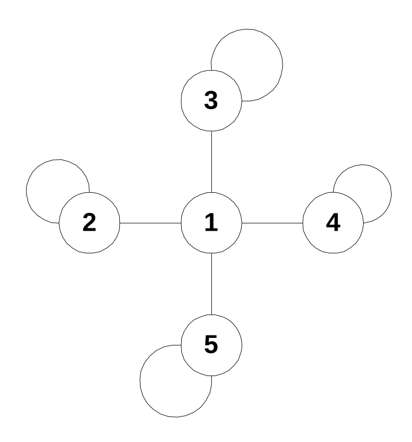
12- **Find the ordered pairs, the domain , and the range of the relation** $y \leq x$, **where x and y are positive integer less than 3.**

13- **For the pair of functions given below, find the *composite function*** $f \circ g$.

f **= {(1, 2), (2, 3), (3, 4), (4, 6)};** ***g*** **= {(a, 1), (b, 2), (c, 4), (d, 3)}**

14- **if** $f(x) = \sqrt[3]{x} - 1$ **and** $g(x) = (x-2)^3 + 1$ **find** $f^{-1}(x)$ **and** $g^{-1}(x)$.

(C)-Write down both the edge set and the *symmetric relation* of the following graph.



15- Let V be the set {{1, 2}, {3, 4}, {1, 2, 3}, {4, 5}, {6, 8}, {7, 8, 9}, {7} }. Defined a relation R on V by $(X, \; Y) \in R \quad if \; X \cap Y \neq \phi$.

(a) Show that R is symmetric, and draw the graph of R.

(b) Write down the connected components of this graph.

16-For each pair of functions below, the image of g **is a subset of the domain of** f**, so that** $f \circ g(x)$ **is defined. Give the best description you can for the composite function** $f \circ g(x)$.

a) f ={(1.-1) (2,-2) (3,-3) (4,-4) (5,-6)} g ={(1,3),(2,4),(3,5)}

17-Given $f = \{(1, a), (2, c), (3, b)\}$ **from** $X = \{1, 2, 3\}$ **to** $Y = \{a, b, c\}$ **is one-to-one function and onto y. Find** f^{-1} .

18- Prove that the following relation R , is reflexive, symmetric, and transitive and show its Diagraph.

$$R = \{(1,1),(1,3),(1,5),(3,1),(3,3),(3,5),(5,1),(5,3),$$
$$(5,5),(2,2),(2,6),(6,2),(6,6),(4,4)\}$$

19- Write down the matrix of the relation R from $X = \{2,3,4\}$ **to** $Y = \{5,6,7,8\}$ **relative to the orderings 2,3,4 and 5,6,7,8, defined by:**

xRy *if x divides y*

الفصل الرابع
انظمة العد
Numbering systems

1-4 تركيب البيانات The Data Hierarchy

ان كل انواع البيانات المُعالجة بواسطة الحاسوب الرقمـي اختصرت اخـيراً الى خلـيط مـن الاصـفار والواحـدات (1s,0s) . وهذا يحدث بسبب ان ذلك يكون سهلاً واقتصادياً لبناء اجهزة الكترونية تعمل بمستويين مستقرين احـداهما يمثـل الصـفر والثاني يمثل الواحد. ومن الجدير بالملاحظة ان الدوال المؤثرة المنفذة بواسطة الحاسوب تشتمل فقط على اغلب التعـاملات الاساسية للصفر والواحد في المعالجة.

ان اصغر وحدة معلومة في الحاسوب هي البت (Bit) والتي تفترض اما القيمة صفر او القيمـة واحـد , وهـي مختصرـ لــ (BinaryDigit), وهو الرقم الذي يفترض ان يكون واحد مـن اثنـين. مـن جانـب اخـر , ان دوائـر الحاسـوب تنفـذ مختلـف معالجات البتتات (مثل فحص قيمة البت 0 او 1, تضبيطها (*Setting the value of a Bit*) اوعكس قيمـة البـت مـن 0 الى 1 او من 1 الى 0. واذا اردنا ان نخزن المعلومات فانه يتم استخدام مجموعة مـن الخانـات الثنائيـة للتعبـير عـن الحـرف وعادة ماتستخدم ثمانية خانات تدعى بايت واحد (One Byte) وهي اصغر وحدة يتم التعامل معها وحسـب الجـدول (4-1) التالي:

Bit	0 or 1	
Byte	8 bits	
Kilobyte	2^{10} byte (1024 bytes)	$1024=32x32=2^5x2^5=2^{10}$
Megabyte	Kilo kilo byte $=2^{20}$ byte	$1024 \times 1024 = 2^{20}$
Gigabyte	Kilo megabyte= 2^{30} byte	$1024 \times 1024 \times 1024 = 2^{30}$
Terabyte	Kilo gigabyte= 2^{40} byte	$1024 \times 1024 \times 1024 \times 1024 = 2^{40}$

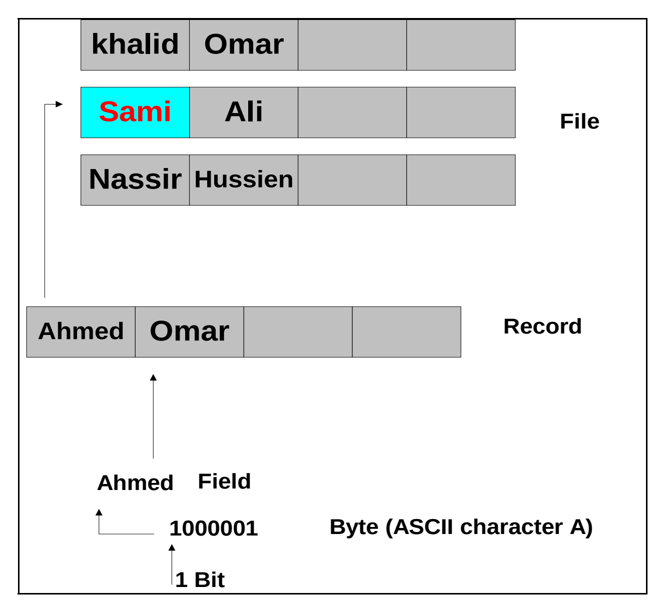
جدول (4-1) وحدات تخزين البيانات

من جانب اخر انه من المتعب للمبرمجين العمل مع البيانات بصيغة المستوى الواطئ للبت . والبديل لذلك هو العمـل مـع البيانات بصيغة الارقام العشرية (0, 1, 2 ....,9)

والحروف الابجدية (من A الى Z او من a الى z) ,والرموز الخاصة مثل (i.e., $, @, %, &, *, (,), -, +, ", :, ?, /, and (many others) . ان الارقام والحروف والرموز تسمى احرف (*Characters*). ان مجموعة كل تلك الحروف تستخدم لكتابة البرامج وتمثيل البيانات في الحواسيب الخاصة وتسمى مجموعة أحرف الحاسوب (*Computer Character Set*)

وبما ان الحاسوب بامكانه ان يعالج 0 و 1 , فان كل حرف من مجموعة احرف الحاسوب يتم تمثيله بسلسلة متتالية من الاصفار والواحدات (0s, 1s) تدعى بايت byte . البايتات (Bytes) بشكل عام تتشكل من ثمانية بتات (8 bits).ان المبرمجون ينشأون البرامج والبيانات بالاحرف ,والحاسوب يعالج ذلك كطراز من البتات (*Pattern of Bits*).

وحيث ان الاحرف تتشكل من البتات فان الحقول (*fields*) تتشكل من الاحرف او البايتات .ان الحقل هو مجموعة من الاحرف التي توصل المعنى . مثلاً الحقل الذي يتضمن حروف وحيدة صغيرة او كبيرة (*Solely of Uppercase And Lowercase Letters*) ممكن يستخدم كأسم شخص .ان انواع البيانات المعالجة بالحاسوب تشكل الشكل الهرمي لتراكيب البيانات (*Data Hierarchy*) والتي فيها تصبح البيانات اكبر واكثر تعقيدا في التراكيب وكما نعالج من البت الى البايت ثم الحقل وهكذا . اما السجل فيدعى بناءاً او صفا في c++ (*in structure or class in c++*)ً ويتكون من عدة حقول تسمى العناصر بلغة C++ . والشكل (1-4) التالي يبين التسلسل المتعلق بتركيب البيانات (*Data Hierarchy*).



The data hierarchy

شكل(1-4) تركيب البيانات

2-4 التمثيل الداخلي للمعلومات : ويتم ذلك بواسطة:

- نظام التشفير الثنائي (*Binary Code System*)
- نظام التشفير الأمريكي القياسي لتبادل المعلومات (American Standard Code for Information Interchange)

يستخدم النظام الثنائي (0, 1) للتعبير عن الرموز المختلفة (كالحروف الابجدية والارقام والعمليات الرياضية +, -, *, ؟ @, # , $, ...الخ) وذلك بجعل عدد معين يمثل ذلك الرمز وهذا مايسمى بنظام التشفير الثنائي Binary (Code System) .

ومن اشهر انظمة التشفير هو نظام التشفير الشهير ASCII (American Standard Code for) Information Interchange)

ان اغلب الطرق الشائعة لتمثيل الاحرف داخلياً في الحاسوب هي بواسطة استخدام طول ثابت لمجموعة البتتات (*Fixed Length Bit Strings*) . على سبيل المثال نظام اسكي (ASCII) يمثل كل حرف بمجموعة من البتتات عددها سبعة . انظر الجدول (4-2) التالي:

character	ASCII code	Decimal
A	100 0001	65
B	100 0010	66
C	100 0100	67
1	011 0001	49
2	011 0010	50
!	010 0001	33
*	010 1010	42

جدول (4-2) نظام اسكي لبعض الرموز

ان الجدول(4-2) ادناه يمثل مجموعة احرف اسكي (ASCII character set) . فالارقام الموجـودة عـلى يسـار الجـدول تمثـل الارقام في اليسار من المكافئ العشري ((0-127) *Decimal Equivalent*). والارقام في اعلى الجدول تمثـل الارقـام في اليمين لشفرة الحرف (*Character Code*) . مثلاً شفرة الحرف F هي 70 وشفرة الرمز & هي 38 وهكذا لبقية مكونات الجدول(4-3) ادناه: [8]

ASCII CHARACTER SET

	0	1	2	3	4	5	6	7	8	9
0	nu1	soh	stx	etx	eot	enq	ack	bel	bs	ht
1	n1	vt	ff	cr	so	si	dle	dc1	dc2	dc3
2	dc4	nak	syn	etb	can	em	sub	esc	fs	gs
3	rs	us	sp	!	"	#	$	%	&	`
4	(	)	*	+	,	-	.	/	0	1
5	2	3	4	5	6	7	8	9	:	;
6	<	=	>	?	@	A	B	C	D	E
7	F	G	H	I	J	K	L	M	N	O
8	P	Q	R	S	T	U	V	W	X	Y
9	Z	[	\	]	^	_	‘	a	b	c
10	d	e	f	g	h	i	j	k	l	m
11	n	o	p	q	r	s	t	u	v	w
12	x	y	z	{	|	}	~	del		

جدول (4-3) رموز نظام اسكي

يلاحظ ان اعلى قيمة في جدول اسكي لاتزيد عن 128 هي 2^7 ,بمعنى اخر اننا نحصل على جميع قيم الرموز في النظام من مجموع بعض الارقام المحصورة بين 1-64 وكمايلي: افرض الحرف A الذي يساوي 65 حيث يمثل بالثنائي ب 1000001 ,وهذه هي   7 bits حيث 65=1+64   انظر الشكل (4-2) التالي:

		2^6	2^5	2^4	2^3	2^2	2^1	2^0
		64	32	16	8	4	2	1
A=65	⇒	1	0	0	0	0	0	1
	Bit no.	7	6	5	4	3	2	1

شكل (4-2) تمثيل الحرف A بنظام الاسكي

ولو اردنا ان نمثل كلمة **hello** باستخدام نظام **ASCII**, فان كل حرف يمثل بـ 8 bits والنتيجة تكون كمايلي:

h	e	l	l	o
104	101	108	108	111
01101000	01100101	01101100	01101100	01101111

شكل (4-3) تمثيل كلمة hello بنظام الاسكي

مثال 4- 1 اختبر نفسك:

اكتب اسمك كما يفهمه الكومبيوتر مستخدما نظام التشفير ASCII للتعبير عنه اذا علمت ان

A=65 ……..……Z=90

a=97…………..z=122 في هذا النظام وان الحرف الاول من الاسم هو حرف كبير (Capital letter)

4-3 انظمة العد Numbering Systems

4-3-1 النظام العشري : Decimal System

4-3-2 النظام الثماني: Octal System

4-3-3 النظام الثنائي : Binary System

4-3-4 النظام السادس عشري : Hexadecimal System

1-3-4 النظام العشري : THE HINDU-ARABIC (DECIMAL) SYSTEM

في كل نظام عددي يوجد ما نسميه برموز النظام(Symbols) وأساس النظام (Base). ان نظام العد هذا هو نظام مرتب مع 10 كأساس له, ويستخدم الرموز المسماة ارقام من (0, 1, 2, 3, 4, 5, 6, 7, 8, 9.) وكل رقم في هذا النظام لـه قيمـة موقعية (*a place value*) . اذن القيمة التي يمثلها اي رقم تعتمد على موقع ذلك الرقم ضمن العدد الـذي يمثلـه . فـالرقم 2 مـثلاً في العـدد 312 تمثـل 2 (واحـدان Two Ones) فقـط بيـنما الـرقم 2 في العـدد 321 تمثـل 20 (عشرتان Two Tens).

$$a^{n} = \underbrace{axaxa \ldots\ldots\ldots a}_{n\ a's}$$

$$a^{0} = 1 \qquad a \neq 0$$

$312 = (3 \times 100) + (1 \times 10) + (2 \times 1)$ -1

$= (3 \times 10^{2}) + (1 \times 10^{1}) + (2 \times 10^{0})$

ان الخطوة الاخيرة تدى صيغة التمدد (*Expanded Form*). **كما ان في النظـام العشري (*Decimal System*) تـم اختيـار الاساس 10 , فمن المحتمل ان يكون بسبب ان الاصابع العشرة ملائمة لعملية العد .**

مثال 4-2:

اكتب الرقم 3406 بصيغة التمدد (*Expanded Form*) واكتب $(3x10^0) + (2x10) + (5x10^3)$ بالصيغة العشرية الاعتيادية (*Ordinary Decimal Form*)

1-Write 3406 in expanded form

$3406 = (3x10^{3}) + (4x10^{3}) + (0x\ 10^{1}) + (6x\ 10^{0})$

2-Write $(5x10^{3}) + (2x10) + (3x10^{0})$ in ordinary decimal form

$(5x10^{3}) + (2x10) + (3x10^{0}) = (5x1000) + (2x10) + 3$

$= 5023$

1- اذن النظام العشري مكون من عشرة رموز مرتبة بحيث كل رمز يمثل عددا اكبر من الرمز الذي على يساره.

ولمعرفة او لتمثيل عدد يلي عدد معين فاننا نضع في خانة الاحاد الرمز الذي يليه في ترتيب الرموز للنظام وهذه **هي القاعدة الاولى** .مثلا مالعدد الذي يلي العدد 65 في النظام العشري ؟

الحل ببساطة هو 66 لان خانة الاحاد تمثل الرقم 5 في العدد 65 , والعدد الذي يلي العدد 5 في النظام العشري هو العدد 6 . اذن حسب القاعدة نبدل الاحاد 5 ب 6 ليصبح الرقم 66 .

2-القاعدة الثانية : اذا كان الرمز هو الاخير كما في 69 مثلا فاننا نضع الرمز الاول في النظام العشري وهو(الصفر) مكانه ثم نبدل الرمز الموجود في الخانة اليسرى له بالرمز الذي يليه في ترتيب النظام .اي هنا الرمز الاخير 9 يبدل ب 0 والذي على يساره وهو الرمز 6 يبدل بالذي يلي 6 وهو 7 ليصبح الرقم 70.

2- ماهو رمز العدد الذي يلي العدد 199 ؟ طبعا العدد 200 لان حسب القاعدة الثانية ان 9 هي الرمز الاخير فيبدل ب 0 والذي على يساره ايضا 9 فلايوجد رقم يليه في النظام فيصبح الاخير ايضا فيبدل ب 0 ايضا ثم ياتي الرقم 1 ليبدل ب 2 حسب القاعدة ليصبح الرقم 200.

مثال 4-3:

الكتب الرقم 10101_2 والرقم $5AC_{sixteen}$ بدلالة النظام العشري

Examples:

1-Write the number 10101_2 in decimal notation

$10101_2=(1x2^4)+ (0x2^3)+(1x2^2)+(0x2^1)+(1x2^0)$

$= 16+4+1=21$

2-Write the number $5AC_{sixteen}$ in decimal notation

$5AC_{sixteen}=(5x16^2)+(10x16^1)+(12x16^0)$

$= (5x256)+160+12 = 1452$

4-4 ارقام الأساس الاخرى :Other Number Bases

من السهل استخدام اي رقم كأساس (Base)لأنظمة العد وكمايلي:

مثال 4-4 :

غيّر العدد 33 الى الاساس 2 والاساس 5

Change the number 33 to : a-Base 2 b-Base 5

a- 33/2 = 16 remain **1**

16/2 = 8 remain **0**

8/2 = 4 remain **0**

4/2 = 2 remain **0**

2/2 = **1** remain **0**

So; $33=100001_2$

33/5 = 6 remain **3**

6/5 = **1** remain **1**

So; $33 = 113_5$

4-4-1 النظام الثنائي: الاساس 2 (Base 2)

هذا النظام يستخدم فقط 1 and 0 . وفي هذا النظام تتشكل الاعداد باستخدام مايسمى (Blocks) التي هي قوى للاساس 2 وكمايلي:

$2^0=1$, $2^1=2$, $2^3=8$, $2^4=16$, ………………

مثال 4-5:

عندما نكتب 1101 في النظام الثنائي فنقول في لغة (نعم \ لا) للحاسوب الاكتروني حيث نعم تمثل 1 ولا تمثل 0 وكمايلي:

block of 8, yes; block of 4, yes; a block of 2 , no; a block of 1, yes .this according to 1 is yes , 0 is no.

$1101_2 = (1 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0)$

$= 8+4+0+1=13$

وهذه بدلالة النظام العشري (*Decimal Notation*) .

مثال 4-6:

اكتب الاعداد من 1 الى 7 بدلالة النظام الثنائي Binary Notation. انظر جدول (4-4)

Write the numbers from 1 to 7 in binary notation

Decimal	Binary
1	1
2	10
3	11
4	100
5	101
6	110
7	111

جدول (4-4) تمثيل الاعداد من 1 الى 7 بدلالة النظام الثنائي Binary Notation

وما هو رمز العدد الذي يلي العدد 101_2 ؟

إن رمز العدد هو 110_2

4-5 التحويل من النظام الثنائي إلى العشري

- **طريقة التحويل تعتمد على أساس النظام.**
- **بإيجاد حاصل جمع مضروب كل رمز في العدد بأساس النظام 2.**
- **نبدأ من الجهة اليمنى ونرفع أول أساس إلى الأس 0، ثم نرفع الأساس الذي يليه إلى الأس 1.**

مثال 4-7 :

حول العدد 101_2 إلى النظام العشري.

$101_2 = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$

$= 4 + 0 + 1 = 5$

حول العدد 1011_2 إلى النظام العشري.

$1011_2 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$

$= 8 + 0 + 2 + 1$

$= 1$

مثال 4-8 :

حوِّل العدد العشري (Decimal Number) 91 الى نظام الثنائي (Binary Notation)

الحل:هو اجراء عمليات متتالية لقسمة العدد 91 على 2 اولاً والاحتفاظ بالباقي ثم اعادة القسمة للرقم التالي 45 وهكذا انظر الجدول(4-5) ادناه ولاحظ مواقع البتتات (Bits)

2	91	Remainder=1	1's bit
2	45	Remainder=1	2's bit
2	22	Remainder=0	3's bit
2	11	Remainder=1	4's bit
2	5	Remainder=1	5's bit
2	2	Remainder=0	6's bit
2	1	Remainder=1	7's bit
	0		

جدول (4-5) اجراء عمليات متتالية لقسمة العدد 91 على2

اذن العدد 91 يساوي بالتمثيل الثنائي 1011011_2

$$91 = 1011011_2$$

هنك طريقة اخرى لايجاد العدد 91 بدلالة النظام الثنائي وهي:

1-نقسم العدد 91 الى مجموع اعداد صحيحة ممثلة بدلالة الاساس 2 . اي ان :

$$91 = 64 + 16 + 8 + 2 + 1$$

او كتابتها بدلالة الاساس 2 كمايلي:

$$91 = 2^6 + 2^4 + 2^3 + 2^1 + 2^0$$

وبمقارنة هذه الارقام التي بدلالة الاساس 2 مع مواقعها بالجدول ادناه نحصل على النتيجة, حيث نمثل الموقع بالبت 1 عندما يكون موجوداَ والرقم غير الموجود نمثلة بالبت 0 , انظر الشكل (4-4).

		2^6	2^5	2^4	2^3	2^2	2^1	2^0
		64	32	16	8	4	2	1
91=	⇒	1	0	1	1	0	1	1
	Bit no.	7	6	5	4	3	2	1

شكل (4-4) تمثيل الرقم 91 بالنظام الثنائي

لاحظ ان العددين 4 و 32 غير موجودين ضمن مجموع الاعداد المكونة للرقم 91 .اذن العدد 91 بالنظام الثنائي هو

$$91 = 1011011_2$$

6-4 اضافة الثنائي :Binary Addition

مثال 4-9:

اضف الاعداد الثنائية 10011011 و 1011011

اولاً نكتب المسألة كمايلي:

```
  10011011
+  1011011
-------------
```

كما هو في اضافة الارقام العشرية فاننا نبدأ من اليمين .نضيف 1 و 1 , $(1+1=10_2)$. هذا الجمع هـو عبـارة عـن 10_2. اذن نكتب 0 ونحمل 1 من الرقم 10_2 , وعند هذه النقطة الحساب يكون:

```
          1
    10011011
+    1011011
-------------
           0
```

في الخطوة القادمة نكمل من اليمين ايضاً و نضيف 1 و 1 و 1 . $((1+1)+1=10+1=11_2)$ وفي هذه الحالة نكتب 1 ونحمل 1 . وعند هذه النقطة يكون الحساب كمايلي:

```
         1
    10011011
+    1011011
-------------
          10
```

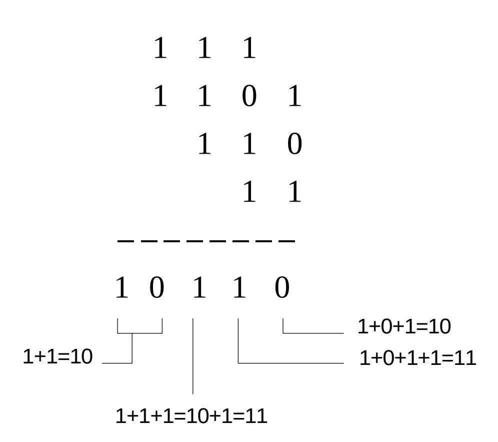
وبالحساب بهذه الطريقة لكل الارقام الثنائية المتبقية نحصل في النهاية على:

```
    10011011
+    1011011
-------------
    11110110
```

مثال 4-10:

اوجد مجموع الاعداد الثنائية التالية: $(1101_2+110_2+11_2)$

سنحذف الاساس 2 اثناء الحساب للاعداد الثنائية اعلاه , لكن نضع في بالنا ان كل الارقام هي ثنائية



وعلى اساس ذلك سيكون المجموع هو :

$$(1101_2 + 110_2 + 11_2 = 10110_2)$$

مثال 4-11:اوجد حاصل طرح : $(1111_2 - 110_2)$

الحل :انظر الشكل ادناه

```
    1 1 1 1
  -   1 1 0
  ---------
    1 0 0 1
```

1 - 0 = 1

1 - 1 = 0

1 - 1 = 0

اذن النتيجة هي:

$$(1111_2 - 110_2 = 1001_2)$$

مثال 4-12:

اوجد حاصل طرح $(1010_2 - 101_2)$

```
   0  10  0  10
      0   1   0
   -  1   0   1
   ______________
0 - 1 = 10 - 1 = 1 ____ 1  0  1 ____ 0 - 1= 10 - 1 = 1
                           |
                       0 - 0 = 0
```

لاحظ اذا بدأنا من اليمين ونطرح $(1-0)$ فان ذلك لايجوز .اذن نحتاج لاعارة (1) من جهة اليسار ليصبح $(10_2 - 1_2 = 1)$, وهكذا نستمر, انظر الشكل اعلاه , وتكون النتيجة النهائية هي:

$(1010_2 - 101_2 = 101_2)$ ويمكننا التحقق من النتيجة وذلك باضافة $(101_2 + 101_2)$ لنحصل على (1010_2).

مثال 4-13:

اوجد حاصل ضرب $(101_2 x110_2)$. تتم عملية الضرب من اليمين وذلك بضرب الرقم 1 بالارقام 110 , ثم ضرب الرقم 0 بنفس الارقام , وهكذا تستمر العملية بنفس عملية الضرب في الحساب ثم الجمع للحصول على الناتج النهائي .

```
      1  1  0
   x  1  0  1
  -------
         1  1  0
      0  0  0
   1  1  0
  ---------
   1  1  1  1  0
```

من الشكل اعلاه ان نتيجة الضرب هي:

$$(101_2 x110_2 = 11110_2$$

مثال 4-14:

اوجد حاصل قسمة (1011_2) على 10_2

$$\begin{array}{r} 1\ 0\ 1 \\ 10\overline{)1\ 0\ 1\ 1} \\ -\ 1\ 0\quad\ \ \\ \hline 1\ 1 \\ -\ 1\ 0 \\ \hline 1 \end{array}$$

1st step

2nd step

remainder

-1 الخطوة الاولى Step 1 : ان $(1x10 = 10)$, اذن نكتب 1 في البسط فوق 10 ثم نطرح.

-2 الخطوة الثانية Step 2 : ننزل الرقم التالي وهو 1 وهو لايقسم على 10 , وبذلك نكتب 0 في البسط ونزل الرقم الاخير 1 ليصبح لدينا الرقم (11) .

-3 نقسم ا الرقم 11على 10 والنتيجة 1 في البسط والباقي 1 بعد الطرح.

اذن نتيجة القسمة هي:

$(1011_2 \div 10_2 = 101_2$ والباقي 1 . ويمكننا التاكد من صحة ذلك من خلال ضرب (101_2) في (10_2) واضافة 1 لنحصل على (1011_2):

$$(101_2 x10_2) + 1 = 1011_2$$

4-7 تمثيل النظام الثماني Octal System Representation

-1 رموز النظام: يتكون هذا النظام من ثمانية رموز هي: 0, 1, 2, 3, 4, 5, 6, 7

■ -2 أساس النظام: 8 , انظر الجدول (4-6)

Binary	Octal
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

الجدول (4-6) تمثيل النظام الثماني

من الجدول اعلاه نشاهد ان الارقام في النظام الثنائي ثلاثة (3 Digits) يقابلها رقم واحد في النظام الثماني

We can see three digit binary numbers correspond to one digit octal number throughout .

مثال 4-15:

1- ما هو رمز العدد ثمانية الذي يلي العدد سبعة والممثل بالرمز 7 في النظام الثماني؟

الرمز 10 هو الذي يمثل العدد ثمانية في النظام الثماني، لأن الرمز 7 هو آخر رمز في النظام الثماني.

مثال 4-:16

2-ما هو رمز العدد الذي يلي العدد 477_8 ؟

ان رمز العدد هو الرمز 500_8

مثال 4-:17

3- حول 11101110_2 الى النظام الثماني

نقسّم الرقم المعطى الى مجاميع مكونة من ثلاثة ارقام كما هو ادناه ونكتب المكافئ لكل مجموعة بالنظام الثماني (الصفر يرفق دائماً الى يسار اغلب المجاميع .انظر الى جدول النظام الثماني وقارن)

Example 3:

Convert 11101110_2 to octal notation.

We break the given number into groups of three and write the octal equivalent of each group (a 0 was attached to left most group): see octal system table and compare

011	101	110
↓	↓	↓
3	5	6

Thus; **$11101110_2=356_8$**

اذن الرقم بدلالة النظام الثماني هو: $11101110_2=356_8$

مثال 18-4:

4- حوّل 5732_8 الى النظام الثاني .

بالمقارنة مع جدول(4-6) النظام الثماني في المثال السابق نحصل على الجدول التالي:

Example4:

Convert 5732_8 to binary notation. Comparing with octal system table ,we get:

5	7	3	2
↓	↓	↓	↓
101	111	011	010

Thus **$5732_8=101111011010_2$**

مثال 19-4: التحويل من النظام الثماني إلى العشري

حول العدد 32_8 إلى العشري.

$32_8 = 3 \times 8^1 + 2 \times 8^0$

$= 24 + 2$

$= 26$

4-8 النظام السادس عشر Hexadecimal System

1- رموز النظام: يتكون هذا النظام من 16 رمزاً هي: 0 ,1 ,2 ,3 ,4 ,5 ,6 ,7 ,8 ,9, F,E,D,C,B,A

2- أساس النظام: **16**

وفي هذا النظام توجد ستة أحرف في نهاية قائمة رموز النظام و تُمثل في النظام العشري . انظر جدول (4-7) كالآتي:

النظام السادس عشر	A	B	C	D	E	F
النظام العشري	**10**	**11**	**12**	**13**	**14**	**15**

جدول (4-7) اساس النظام السادس عشر مقارنة مع النظام العشري

والجدول (4-8) التالي لتمثيل النظام السادس عشر بالنظام الثنائي :

Binary	Hexadecimal
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

الجدول (4-8) التالي لتمثيل النظام السادس عشر بالنظام الثنائي

من الجدول(4-8) اعلاه نشاهد ان الارقام في النظام الثنائي اربعة (4Digits) يقابلها رقم واحد في النظام السادس عشر

We can see **four digit binary numbers** correspond to **one digit hexadecimal number** throughout .

مثال 4-20 :

1-ما هو رمز العدد الذي يلي العدد خمسة عشر والممثل بالرمز F_{16}؟

إن رمز العدد ستة عشر بالنظام السادس عشري ممثل بالرموز 10_{16}

2-ما هو رمز العدد الذي يلي العدد الممثل بالرمز CF_{16}؟

إن رمز العدد يكون $D0_{16}$.

مثال 4-21: التحويل من النظام السادس عشري إلى العشري

1- حول العدد 15_{16} إلى العشري.

$15_{16} = 1 \times 16^1 + 5 \times 16^0$

$= 16 + 5$

$= 21$

2- حول العدد $3D_{16}$ إلى العشري.

$3D_{16} = 3 \times 16^1 + 13 \times 16^0$

$= 48 + 13$

$= 61$

3- حوّل 1101110_2 بدلالة النظام السادس عشر Hexadecimal Notation

طبقاً الى جدول النظام السادس عشر , فاننا نقسّم ارقام العدد الى مجموعتين في كل مجموعة اربعة ارقام كما هو في الشكل ادناه

Example 3: Convert 1101110_2 to hexadecimal notation

According to the hexadecimal table ,we divide the digits in the binary numeral into group of four:

0110	1110
↓	↓
6	14=E

So that 1101110_2=$6E_{16}$

4- حوّل $9AD_{16}$ بدلالة النظام الثنائي Binary Notation

Example: Convert $9AD_{16}$ to binary notation.

9	A	D
↓	↓	↓
1001	1010	1101

Thus , $9AD_{16}$=100110101101_2

5- - حوّل $9AD_{16}$ بدلالة النظام الثماني Octal Notation

Example: We have $9AD_{16}$=100110101101_2 .converting this binary numeral to octal

بما انه نريد التحويل الى النظام الثماني . اذن نقسّم الرقم الثنائي الى مجاميع كما ذكرنا انفاً , في كل مجموعة 3 أرقام وكما في الجدول ادناه:

can be done as follows: because we want to convert to octal, so we divide the binary into group of 3.

100	110	101	101
↓	↓	↓	↓
4	6	5	5

Thus , $100110101101_2=4655_8$, so that $9AD_{16}=4655_8$

مثال 4-22 للحل:

Homework: Convert 357_8 to hexadecimal notation.

4-8-1 الاضافة في النظام السادس عشر Hexadecimal Addition

مثال 4-23:

اضف اعداد النظام السادس عشر $(84F_{16})$ و $(42EA_{16})$

يمكننا كتابة المسألة بالشكل التالي:

```
   84F
+ 42EA
--------------
```

نبدأ من اليمين ونضيف F الى A وبما ان F هي (15_{10}) و A هي (10_{10}) ، اذن حاصل جمعهما هو:

$$(F + A = 15_{10} + 10_{10} = 25_{10} = 19_{16})$$

وبذلك نكتب 9 ونحمل 1. كما في الشكل ادناه:

```
     1
   84F
+ 42EA
-----------
     9
```

في الخطوة القادمة : نضيف 1, 4 , و E . وبنفس الطريقة نكتب 3 ونحمل 1 كما في الشكل ادناه:

```
   1
  84F
+ 42EA
-----------
   39
```

ونستمر بنفس الطريقة لنحصل على:

```
  84F
+ 42EA
-----------
  4B39
```

وبذلك تكون نتيجة الجمع هي:

$$(84F_{16} + 42EA_{16} = 4B39_{16})$$

وفيما يلي برنامج بلغة البيسك لتحويل الاعداد في اي نظام ولأساس اقل من 17 الى مايقابلها في اي نظام اخر ولأساس اقل من 17 ايضاً [3]:

```
THIS PROGRAM TO CHANGE FROM BASE X TO BASE Y(X, Y<17)

10 DIM A$(16): FOR I = 0 TO 9: A$(I) = RIGHT$(STR$(I), 1): NEXT
20 A$(10) = "A": A$(11) = "B": A$(12) = "C": A$(13) = "D": A$(14) = "E": A$(15)= "F": A$(16)= "G":
30 CLS : PRINT "THIS PROGRAM TO CHANGE FROM BASE X TO BASE Y(X, Y<17)"
35 PRINT
36 PRINT
40 INPUT "ENTER THE ORIGINAL BASE OF THE NUMBER"; B
```

```
50 INPUT "ENTER NUMBER IN THE ORIGINAL FORM PLS"; X$: A$ = X$: K = LEN(X$): S$=X$
60 FOR I = 1 TO K
70 S$ = RIGHT$(X$, 1): X$ = LEFT$(X$, K - I)
80 S = VAL(S$): FOR J = 10 TO 16: IF S$ = A$(J) THEN S = J
90 NEXT J
100 T = S * B ^ (I - 1) + T
110 NEXT I
120 N = T: INPUT "CHANGE TO WHAT BASE"; K: X$ = " "
130 L = INT(N / K): R = N - L * K: N = L
140 IF L < K THEN GOTO 160
150 X$ = RIGHT$(STR$(R), 1) + X$: GOTO 130
160 X$ = A$(L) + A$(R) + X$
170 PRINT "THUS , "; A$; " IN BASE "; B; "="; X$; " IN BASE"; K
180 PRINT : INPUT "RUN AGAIN (Y/N)"; A$: IF A$ = "Y" THEN RUN
190 END
```

مثال على تنفيذ البرنامج

```
THIS PROGRAM TO CHANGE FROM BASE X TO BASE Y(X, Y<17)


ENTER THE ORIGINAL BASE OF THE NUMBER?    2
ENTER NUMBER IN THE ORIGINAL FORM PLS?    11101110
CHANGE TO WHAT BASE?  8
THUS, 11101110 IN BASE 2 = 356 IN BASE 8
```

9-4 نظام العد للاساس 4 :: Base 4 System of Numeration

ان ارقام الاساس 4 هي 0, 1, 2, و 3 . وكل هذه الارقام يمكن ان تتشكل من مجموع الارقام 0, 1, و 2 . ويمكننا ان ننشأ جدولاً لنظام العد بالاساس 4 مشابه لما عملناه في النظام الثماني او السادس عشر.....الخ . انظر الجدول(4-9) ادناه:

Binary	Base 4
00	**0**
01	**1**
10	**2**
11	**3**

جدول (4-9) تمثيل نظام العد للاساس 4 بالنظام الثنائي

مثال 4-24:

حوّل 11110_2 الى نظام الاساس 4 :

Example: Convert 11110_2 to base 4.

01	11	10
↓	↓	↓
1	3	2

This shows that $11110_2=132_4$

تمارين الفصل الرابع

1- Show how to change 312_4 **to binary notation**

2- Convert 203_4 **to binary notation**

3- Convert 312_4 **to hexadecimal notation**

4- Convert 1101111_2 **to base 4 notation**

5- Use the program at the end of this chapter to change from base x to base y(x, y<17).

6- Add 1111_2 **to** 1101_2

7- Do the following the following computations in the hexadecimal system:

$2BC_{16} + 5D_{16}$; $3C4_{16}\, X\, 2B_{16}$

الفصل الخامس

الجبر البولياني

Boolean Algebra

تعريف (5-1) :

الجبر البولياني يزودنا بالخصائص الجبرية لعمليات المجموعة التي تجعلنا نتحقق من صحة المعادلات التي تشتمل على مجموعات وبشكل كبير بنفس الطرق التي نستخدمها في الجبر الاعتيادي ordinary algebra الذي يجعلنا نتحقق من صحة المعادلات التي تشتمل على الثوابت (الاعداد) والمتغيرات variables .

Boolean algebra provides us with algebraic properties of set operations that let us verify equation involving sets in much the same ways that ordinary algebra lets us verify equations involving numbers and variables

يقسّم الجبر البولياني الى قسمين :

1- الجبر البولياني للمجموعات Boolean Algebra for sets

2- الجبر البولياني للجمل Boolean Algebra for statements

ففيما يتعلق بالجبر البولياني للمجموعات اننا رأينا انه لبيان ان جملتين حول الكون هما متكافئتان , ان ذلك يجب في بعض الاحيان ان نبين ان المجموعتان متساويتين , وقد راينا ايضاً ان البراهين تستخدم المبادئ الاساسية لتساوي المجموعات لبيان ان المجموعتين متساويتين اصبحت مستهلكة . ان الجبر البولياني يزودنا بالخصائص الجبرية لعمليات المجموعات وكما ذكرناها في موضوع المجموعات وخصائصها . راجع الفقرة وسنذكر بعضاً منها هنا :

نظرية (5-1):

ان عمليات المجموعة على المجموعات الجزئية P, Q و R من المجموعة الكونية (U (Universal تحقق القوانين الجبرية التالية:

Theorem 4: The set operations on subsets of a universe U satisfy the following algebraic laws:

$P \cup (Q \cup R) = (P \cup Q) \cup R$ $P \cap (Q \cap R) = (P \cap Q) \cap R$	(Commutative laws)	(1)
$P \cup Q = Q \cup P$ $P \cap Q = Q \cap P$	(Commutative laws)	(2)
$P \cap (Q \cup R) = (P \cap Q) \cup (P \cap R)$ $P \cup (Q \cap R) = (P \cup Q) \cap (P \cup R)$	(Distributive laws)	(3)
$P \cap (P \cup Q) = P$ $P \cup (P \cap Q) = P$	(Absorptive laws)	(4)
$P \cup \phi = P$ $P \cap U = P$	(Identity laws)	(5)
$P \cup \sim P = U$ $P \cap \sim P = \phi$	(Inverse laws)	(6)

ان برهان اي قانون اعلاه يمكن باستخدام المبدأ الذي " مثلاً تكون مجموعتان متساويتان اذا كان كل عنصر في احداهما هو عنصراً للاخرى , وهلم جراً للقوانين الاخرى ". كذلك يمكن برهنة اي قانون من خلال ترجمته الى المعادلات المطابقة له المشتملة على الرموز ($\vee, \wedge, and \neg$) للجمل. فمثلا ترجمة القانون الاول والثاني اعلاه يعطينا جمل مكافئة و يمكّننا بالتالي من برهنتهما بواسطة جداول القيم (*Truth tables*).

عندما نستخدم القوانين المعروفة لعمليات المجموعة لأشتقاق قوانين جبرية اخرى فاننا نقول اننا نعمل جبراً بوليانياً (Boolean Algebra) . انه الاستخدام للقوانين التي تخص الجبر البولياني وليس اننا نعمل مع المجموعات.

مثال 5-1:

استخدم الجبر البولياني لأثبات خاصية التقاطع الخالية (*empty intersection property*) التالية:

$$\phi \cap P = \phi.$$

الحل:

نكتب المعادلات التالية لنصل الى الاثبات :

$$\phi \cap P = \phi \cap (P \cup \phi) \quad \text{(Law (5) (identity law))}$$
$$= \phi \cap (\phi \cup P) \quad \text{(Law (2) (commutative law))}$$
$$= \phi \quad \text{(Law (4) (absorptive law))}$$

مثال 5-2:

استخدم الجبر البولياني لأثبات خاصية التماثل الوحيـدة(***unique identity property***) , اذا كانـت $X \cap P = P$ لكـل المجموعات الجزئية P لـ U , فان $X = U$

الحل:

استخدام $P = U$, يمكننا كتابة :

$$X \cap U = U$$

$$X = U \qquad \text{(Identity law)}$$

مثال 5-3:

استخدم الجبر البولياني لأثبات خاصية التماثل الوحيـدة(***unique identity property***), اذا كانـت $X \cap P = \phi$ و $X \cup P = U$, فان $X = \sim P$

الحل:

$$X = X \cap U$$
$$= X \cap (P \cup \sim P) \quad \text{(Inverse law)}$$
$$= (X \cap P) \cup (X \cap \sim P) \quad \text{(Distributive law)}$$
$$= \phi \cup (X \cap \sim P) \quad \text{(Given property)}$$
$$= (P \cap \sim P) \cup (X \cap \sim P) \quad \text{(Inverse law)}$$
$$= (P \cup X) \cap \sim P \quad \text{(Distributive law)}$$
$$= U \cap \sim P \quad \text{(Given property)}$$
$$= \sim P \quad \text{(Identity law)}$$

مثال 5-4:

استخدم الجبر البولياني لأثبات قانون النفي المضاعف التالي:

Use Boolean algebra to prove the *double negation law*: ~ ~ Q=Q.

الحل:

استخدام المثال(5-3) بـ $P =\sim Q$ ، فانه يقال انه اذا ($X \cap \sim Q = \phi$) و $X \cup \sim Q = U$ () , فانه $X =\sim (\sim Q)$. مع ذلك بواسطة القانون العكسي (Inverse law) نعرف ان ($Q \cap \sim Q = \phi$) وان $Q \cup \sim Q = U$ ().وطبقاً الى المثال اعلاه , فان المجموعة X فقط حيث ان $X \cup \sim Q = U.$) و ($X \cap \sim Q = \phi$) هي مساوية الى ((~Q) ~) , وبذلك فان (~Q) ~=Q.

5-2 الجبر البولياني للجمل Boolean Algebra for statements

القانون النموذجي للجبر البولياني للمجموعات هو :

$$P \cap (Q \cap R) = (P \cap Q) \cap R,$$ قانون الربط (Associative Law .)

وفي هذا القانون لو فرضنا ان P تقوم مقام المجموعة الحقيقية لـلجملة p و Q تقوم مقام المجموعة الحقيقية لـلجملة q , و R تقوم مقام المجموعة الحقيقية لـلجملة r , فان $P \cap (Q \cap R)$ تقوم مقام الجملة $p \wedge (q \wedge r)$ و $(P \cap Q) \cap R$ تقوم مقام $(p \wedge q) \wedge r$. ونرى ان جملتين تكونان متكافئتين اذا كانت المجموعتان الحقيقيتان لهما متساويتين . وعلى اساس ذلك ان صيغة تقول ان مجموعتان متساويتان طبقاً الى صيغة تقول ان جملتين متكافئتين, لهذا فان :

$$P \cap (Q \cap R) = (P \cap Q) \cap R$$

تترجم الى الصيغة

$$p \wedge (q \wedge r) \Leftrightarrow (p \wedge q) \wedge r$$

لقد راينا القوانين المتعلقة بالجبر البولياني للمجموعات وانه بالامكان تحويلها الى مايتعلق بالجمل بشكل مباشر ماعدا القانون المعكوس (reverse law) وقانون التماثل (identity law) . وحيث راينا ان القنون المعكوس وقانون التماثل في حالة المجموعات يحويان الرمزين U و ϕ اللذين يقومان مقام المجموعة الكونية والمجموعة الخالية على التوالي. فانه لحد الان لاتوجد لدينا رموز مشابهة في حالة الجمل .

اذن لانستطيع تحويل هذين القانونين . لذا فانه من التقليد ان نضع الرقم (1) بشكل غامق للدلالة على بعض الجمل التي مجموعاتها الحقيقية هي U مثل $(p \vee \neg p)$, ونضع (0) غامق للدلالة على بعض الجمل التي مجموعاتها الحقيقية هي ϕ مثل $p \wedge \neg p$, ومن ثم كل القوانين الستة اعلاه يمكن ترجمتها الى مايخص الجمل .

ان الجمل حول الكون تعتبر متكافئات مهمة وغالباً ماتسمى المتماثلات المنطقية وتحقق القواعد التالية:

The statements about a universe satisfies the following rules. ***These important equivalence , often called logical identities***

$(p \vee q) \vee r \Leftrightarrow p \vee (q \vee r)$ $(p \wedge q) \wedge r \Leftrightarrow p \wedge (q \wedge r)$	(Associative laws of $\vee$)
$p \vee q \Leftrightarrow q \vee p$ $p \wedge q \Leftrightarrow q \wedge p$	(Commutative laws)
$p \wedge (q \vee r) \Leftrightarrow (p \wedge q) \vee (p \wedge r)$	(Distributive laws of $\wedge$ over $\vee$)
$p \vee (q \wedge r) \Leftrightarrow (p \vee q) \wedge (p \vee r)$	(Distributive laws of $\vee$ over $\wedge$)
$p \wedge (p \vee q) \Leftrightarrow p$ $p \vee (p \wedge q) \Leftrightarrow p$	(Absorptive laws)
$p \vee 0 \Leftrightarrow p$ $p \wedge 1 \Leftrightarrow p$	(Identity laws)
$p \vee \neg p \Leftrightarrow 1$ $p \wedge \neg p \Leftrightarrow 0$	(Inverse laws)

ان القوانين العكسية اعلاه (*Inverse Laws*) تمثل الجملة الصحيحة (*true statement*) او مايسمى بـ (*Tautology*) والجملة النقيضة (*false statement*) او مايسمى بـ (*Contradiction*).، على التوالي.

ان اية حقيقة يمكن اشتقاقها من قوانين الجبر البولياني المتعلق بالمجموعات يمكننا بشكل مشابه ترجمتها الى حقائق حول الجمل . وعلى اساس ذلك فان لدينا الحقائق التالية:

$$\neg(p \vee q) \Leftrightarrow \neg p \wedge \neg q$$ (DeMorgan's laws)

$$\neg(p \wedge q) \Leftrightarrow \neg p \vee \neg q$$

$$\neg\neg p \Leftrightarrow p$$ (Double negation law)

$$p \vee p \Leftrightarrow p$$ (Idempotent properties)

$$p \wedge p \Leftrightarrow p$$

$$0 \wedge p \Leftrightarrow 0$$ (Zero property)

$$1 \vee p \Leftrightarrow 1$$ (One property)

مثال 5-5: اعد بصيغة مبسطة كتابة التعبير التالي:

$$(\neg\neg \mathrm{r}) \vee (\mathrm{s} \vee (\mathrm{r} \wedge \mathrm{t}))$$

الحل:

في كل سطر ادناه استعملنا قاعدة واحدة من الجبر البولياني لجعل التعبير اعلاة بصيغة مبسطة عمًا بدأنا به وكمايلي:

$$(\neg\neg r) \vee [s \vee (r \wedge t)] \Leftrightarrow r \vee [s \vee (r \wedge t)]$$ (Double negation)

$$\Leftrightarrow [s \vee (r \wedge t)] \vee r$$ (Commutative law)

$$\Leftrightarrow s \vee [(r \wedge t) \vee r]$$ (Associative law)

$$\Leftrightarrow s \vee r$$ (Absorptive law)

مثال 5-6: اعد بصيغة مبسطة كتابة التعبير التالي:

$$(\neg(r \wedge s) \vee (r \vee s)) \wedge (\neg(r \vee s) \vee (r \wedge s))$$

الحل:

في كل سطر ادناه استعملنا قاعدة واحدة من الجبر البوليايي لجعل التعبير اعلاة بصيغة مبسطة عمًا بدأنا به وكمايلي:

$$(\neg(r \wedge s) \vee (r \vee s)) \wedge (\neg(r \vee s) \vee (r \wedge s))$$

$\Leftrightarrow [(\neg r \vee \neg s) \vee (r \vee s)] \wedge [\neg(r \vee s) \vee (r \wedge s)]$	(Demorgan's law)
$\Leftrightarrow [(\neg s \vee \neg r) \vee (r \vee s)] \wedge [\neg(r \vee s) \vee (r \wedge s)]$	(Commutative law)
$\Leftrightarrow [(\neg s) \vee (\neg r \vee (r \vee s))] \wedge [\neg(r \vee s) \vee (r \wedge s)]$	(Associative law)
$\Leftrightarrow [(\neg s) \vee (\neg r \vee r) \vee s] \wedge [\neg(r \vee s) \vee (r \wedge s)]$	(Associative law)
$\Leftrightarrow [(\neg s) \vee (1 \vee s)] \wedge [\neg(r \vee s) \vee (r \wedge s)]$	(Inverse law)
$\Leftrightarrow [(\neg s) \vee 1] \wedge [\neg(r \vee s) \vee (r \wedge s)]$	(One property)
$\Leftrightarrow 1 \wedge [\neg(r \vee s) \vee (r \wedge s)]$	(One property)
$\Leftrightarrow \neg(r \vee s) \vee (r \wedge s)$	(Identity property)
$\Leftrightarrow (\neg r \wedge \neg s) \vee (r \wedge s)$	(Demorgan's law)

3-5 الدوائر الالكترونية لأختبار الجمل

Circuits To Test The Truth Of Statements

في الحواسيب الرقمية (*Digital computers*) هناك امكانيتين , تكتب **0** و **1** , لأصغر شئ غير قابل للتجزئة (*Indivisible Object*) . كل البرامج والبيانات في الاخر تحول الى جمع من البتتات (*Combination of Bits*) , ومختلف الاجهزة التي استخدمت خلال سنوات في الحواسيب الرقمية هي لخزن تلك البتتات . والدوائر الاكترونية سمحت لاجهزة الخزن تلك من الاتصال بعضها ببعضها الاخر .

ان البت في جزء من الدائرة ينتقل الى جزء اخر فيها كفولتية . وعلى اساس ذلك فاننا نحتاج الى مستويين من الفولتية (على سبيل المثال: فولتية عالية تمثل (**1**) وفولتية واطئة تمثل (**0**)). وفي هذا الجزء اننا سنناقش الدوائر المكونة من اكثر من دائرة منطقية (*Combinatorial Circuits*) حيث القيمة الخارجة من اي من تلك الدوائر تعرُف بشكل قيمة وحيدة لكل مزج لمدخلاتها (*inputs*) . ان الدوائر المنطقية المذكورة اعلاه يمكن بناءها باستخدام اجهزة الحالة الصلبة (*Solid State Devices*) التي تسمى **Gates** والتي هي ممكنة لفتح مستويات الفولتية (البتتات Bits).

اذن الحواسيب الرقمية (*Digital computers*) تحوي دوائر الكترونية التي فيها مرور التيار الكهربائي ينظم بواسطة مايسمى (*Gates*) . تلك الدوائر مسؤولة عن الفولتية العالية والواطئة ويمكن وصفها بواسطة جمل منطقية (*Logic Statements*). وان تلك الدوائر تسمى بدوائر الحاسوب او دوائر الكترونية تعمل عمل المفتاح الكهربائي (*Switching Circuits*) وهي كما يلي:

1- دوائر الربط على التوالي وهي (AND gates)

2- دوائر الربط على التوازي وهي(OR gates)

3- دوائر التحويل العكسي (Inverter gates)

في هذه الدوائر الكترونية,على سبيل المثال, المفتاح لعمل رنات الهاتف(*Switch for telephone ring*) , او ميكانيكية نظام تغيير القنوات (*Channel Changing Mechanism*), انه بالامكان غلق او فتح تلك الدوائر الاكترونية بواسطة اشارات كهربائية(*Electrical Signals*).

اما من جهة الحاسوب فانه يمثل صحة وخطأ (*Truth Or Falsity*) للجمل (*Statements*) بواسطة اعلى فولتية (قيمة *1*) او اوطأ فولتية (قيمة **0**) في خلايا الذاكرة (*Memory Cells*) (لان تلك الدوائر ليس فيها ذاكرة) او باسلاك تمثل الجملة . ان استخدام تللك الفولتيات للتحكم في المفاتيح (*Switches*) يسمح للحاسوب باختبار صحة وخطأ الجمل المركبة (*Compound Statements*). فعل سبيل المثال : الدائرة اللازمة لأختبار قيمة الجملة ($p \wedge q$) يجب ان تحوي سلك لادخال قيمة p , وسلك لادخال قيمة q , واخر لاخراج فولتية عالية اذا كانت كل من p و q يمثل فولتية عالية (*1*) . او لاخراج فولتية واطئة ماعدا ذلك. اننا نرى تلك الدائرة وكانها سلسلة من المفاتيح المربوطة والتي تتحكم فيها قيمة الجملة ($p \wedge q$), والجدول (5-1) التالي يسمى جدول الفولتية والذي يمثل كل المدخلات الممكنة من الجمل ($p \wedge q$).

p	q	$p \wedge q$
1	1	1
1	0	0
0	1	0
0	0	0

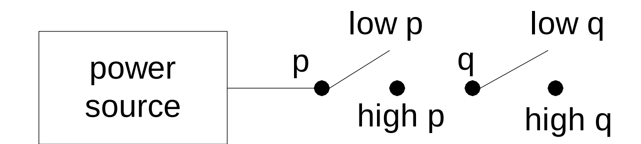
جدول (5-1) جدول الفولتية والذي يمثل كل المدخلات الممكنة من الجمل ($p \wedge q$).

ان الدائرة (AND gate) ادناه تسـتلم مـدخلات (*Inputs*) هـي p و q حيـث ان p و q هـما بـت (bits) وينتجـون مخرجات (*Outputs*) يرمز لهما بـ $p \wedge q$, شكل (5-1) حيث ان :

شكل(5-1) الدائرة (AND gate)

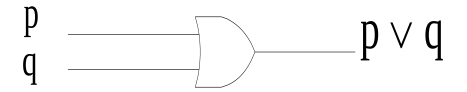
$$p \wedge q = \begin{cases} 1 & \text{if} \quad p = 1 \quad \text{and} \quad q = 1 \quad \text{or} \quad \text{True} \\ 0 & \text{otherwise} \end{cases}$$

ان الدائرة (AND gate) **تمثل ربط على التوالي لمفتاحين** في دائرة كهربائية وتمثل بالشكل (5-2) كمايلي:



شكل(5-2) الدائرة (AND gate) **وتمثل ربط على التوالي لمفتاحين**

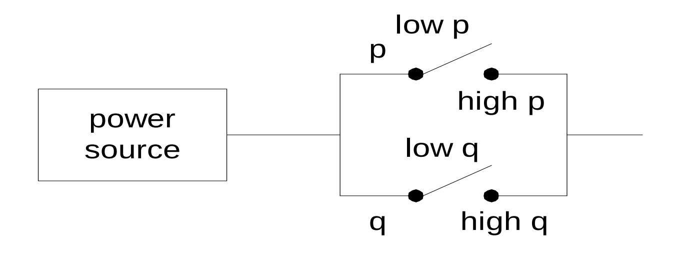
وان الدائرة (OR gate) المبينة ادناه تستلم مدخلات *Inputs*) هي : p و q حيث ان p و q هما بت (Bits) وينتجون مخرجات (*Outputs*) يرمز لهما بـ $p \vee q$ حيث ان :, شكل (5-3) حيث ان :



شكل (5-3) الدائرة (OR gate)

$$p \vee q = \begin{cases} 1 & \text{if} \quad p = 1 \quad \text{or} \quad q = 1 \quad \text{or} \quad \text{true} \\ 0 & \text{otherwise} \end{cases}$$

ان الدائرة (OR gate) تمثل **ربط على التوازي لمفتاحين** في دائرة كهربائية و تمثل بالشكل (4-5) كمايلي:



شكل (4-5) الدائرة (OR gate) وتمثل **ربط على التوازي لمفتاحين**

والجدول (2-5) التالي يسمى جدول الفولتية والذي يمثل نتيجة كل المدخلات الممكنة من($p \vee q$).

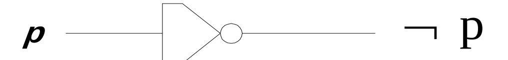
p	q	$p \vee q$
1	1	1
1	0	1
0	1	1
0	0	0

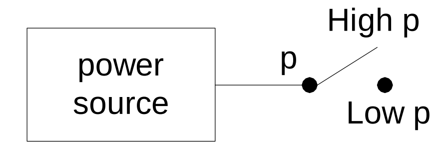
جدول (2-5) جدول الفولتية والذي يمثل نتيجة كل المدخلات الممكنة من($p \vee q$)

ومن الجدول (5-2) نرى ان الدائرة هذه تعطي فولتية عالية في حالة كانت كـل مـن p و q كلاهـما (true) او (احـدهما (*true*) والاخرى (*false*)). وتعطي فولتية واطئة اذا كان كلاهما (*false*) . انظر جدول الفولتية اعلاه.

اما الدائرة ((NOT gate (or Inverter المبينة ادناه فانها تستلم قيمة مدخلة (*Input*) : p حيـث ان p هـي بـت (Bit) وينتج مخرجة (*Output*) يرمز لهما بـ $\neg p$ or $\overline{p}$ حيث ان :

(تعني النفي) شكل (5-5)حيث ان :





شكل (5-5) الدائرة ((NOT gate (or Inverter)

وجدول الفولتية (5-3) كمايلي:

p	¬p
1	0
0	1

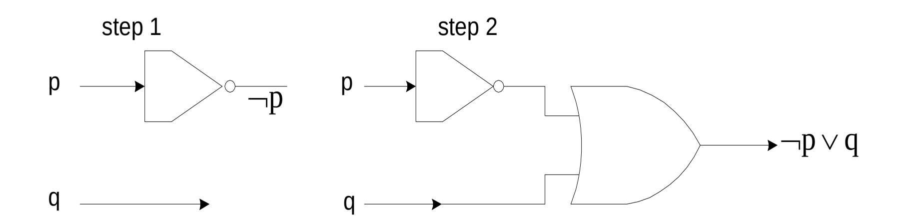
جدول الفولتية (5-3)) للدائرة ((NOT gate (or Inverter)

ومن خلاله نرى ان هذه الدائرة تعطي فولتية عالية عندما p هي خاطئة (*false*) , وتعطي فولتية واطئة عندما p هي (*true*).

*إذن يمكننا ربط هذه السويجات مع بعضها للحصول على دوائر منطقية كثيره في الحاسوب دون الحاجة إلى معرفة في الكهربائية كما في ألامثله القادمة, وهذه الدوائر تسمى (*Combinatorial Circuits*) .

4-5 بناء الدوائر الاكترونية :Constructing Circuits

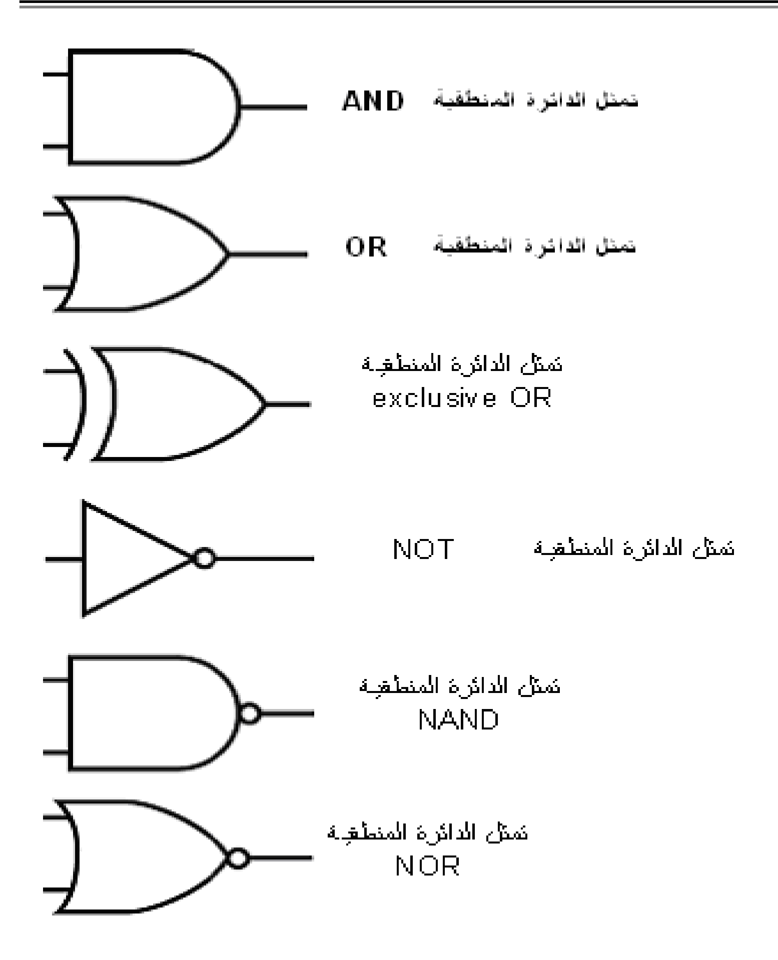
نستخدم في الدائرة المنطقية اللازمة لاختبار الجملة $(\neg p \vee q)$ فيما لو كانت صحيحة (true) , الرموز التي تدل على (AND gate , OR gate, و Inverter gate) شكل(6-5) وكمايلي:



شكل (6-5) *a Combinatorial Network*

ان الشكل الناتج يسمى (*a Combinatorial Network*) بسبب ان ذلك يسمح لنا بفحص القيمة الحقيقية (truth value) للجملة المكونة من ارتباط جمل اخرى (جملة مركبة) . من جانب اخر ان الجملة $p \to q$ () مكافئة للجملة $(\neg p \vee q)$, لـــــذا مــــــن المفيـــــــد ان نعــــــبر عـــــن الجملـــــــة ($p \to q$) بلالة الرموز ($\neg$, $\vee$, $\wedge$).

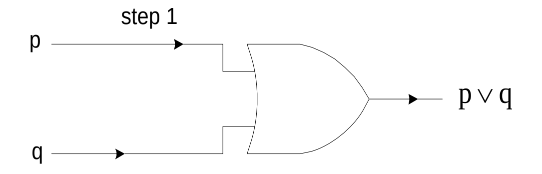
وفيما يلي رموز بعض الدوائرالمنطقية :

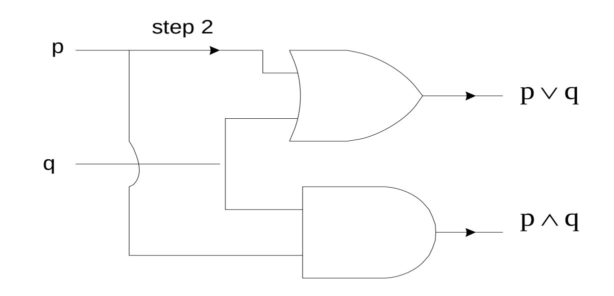


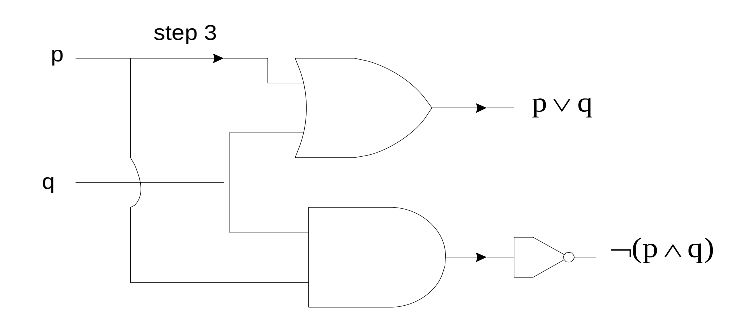
مثال 5-7: بين كيف يمكنك بناء الدائرة اللازمة لفحص القيم الحقيقية للجمل المتكافئة التالية:

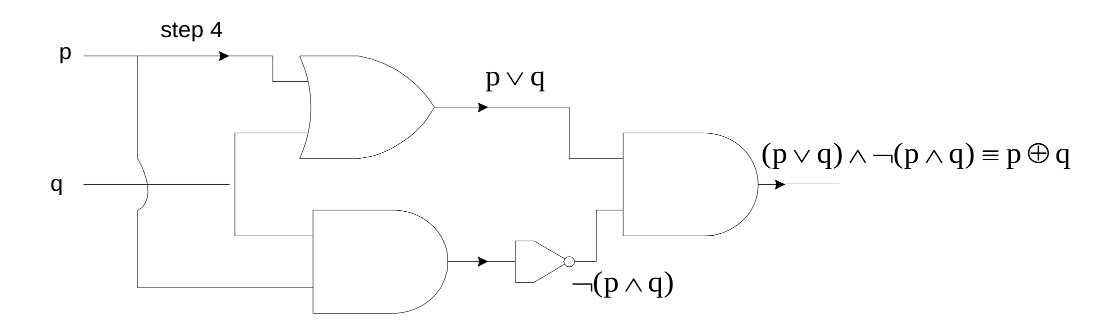
$$p \oplus q \Leftrightarrow (p \vee q) \wedge \neg(p \wedge q)$$

انظر الشكل (5-7) الذي يوضح مراحل بناء الدوائر المنطقية لتحقيق الجملة في هذا المثال







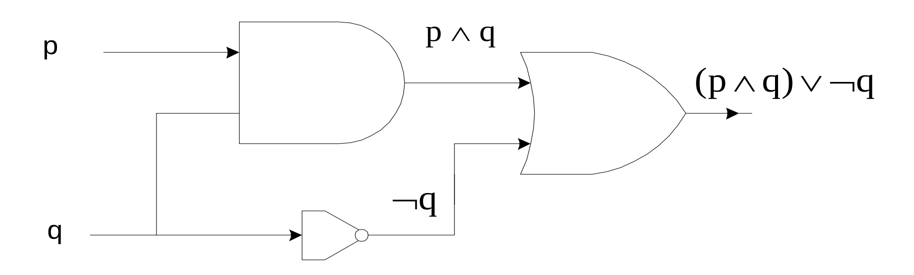


شكل(5- 7) الدائرة الالكترونية لفحص قيم $p \oplus q \Leftrightarrow (p \vee q) \wedge \neg(p \wedge q)$

مثال 5-8:

لكل الاحتمالات المدخلة الممكنة للدائرة التالية شكل(5-8) :

1- اوجد القيمة المخرجة في A
2- اوجد القيمة المخرجة في B
3- اوجد القيمة المخرجة في C



شكل(5- 8) الدائرة الالكترونية لتمثيل (AND gate , OR gate, و Inverter gate)

الحل:

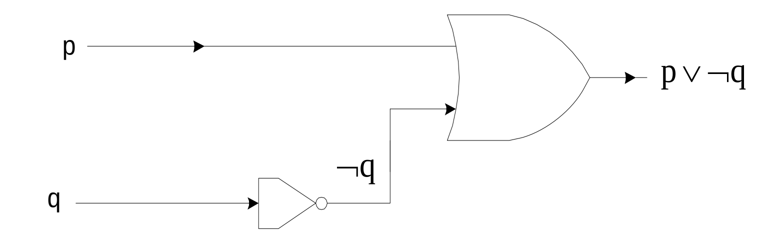
1- القيمة المخرجة في A يرمز لها بـ $p \wedge q$ ()
2- القيمة المخرجة في B يرمز لها بـ ($\sim q$)
3- القيمة المخرجة في C هي نسبة الى $p \wedge q$ و $\sim q$ وبما ان C تمثل (**OR gate**). اذن النتيجة النهائية نرمز لها بـ ($p \wedge q \vee \neg q$).
4- ان القيم المدخلة للجمل p و q هي كما في الجدول (5-4) التالي:

p	q	$p \wedge q$	$\vee$	$\neg q$
1	1	1	1	0
1	0	0	1	1
0	1	0	0	0
0	0	0	1	1

جدول (5-4) القيم المدخلة للجمل p و q والمخرجة نتيجة ربطهما

وبما ان الجملتين التاليتين متكافئتين : $p \wedge q \vee \neg q \Leftrightarrow p \vee \neg q$

فانه من السهولة ان يعبّر عن الدائرة المنطقية اعلاه بالدائرة التالية شكل (5-9) وهي للجملة المكافئة $p \vee \neg q$



شكل(5-9) الدائرة المنطقية للجملة المكافئة $p \vee \neg q$

تمارين الفصل الخامس

1- Show the **logic diagram** and discover the final out put for all possible input for the following statement :

a- $(p \wedge q) \vee (p \wedge r)$ b- $p \vee (q \wedge r)$

2- Show the logic diagrams and discovers the ***final out-puts*** for all possible ***inputs*** for the following statements:

1- $\neg p \wedge \neg q$ 2- $p \vee (q \wedge r)$

3- بين مخطط الدوائر المنطقية للجمل التالية واوجد المخرجات النهائية لكل الامكانيات المدخلة لها:

a- $(p \vee q) \wedge (\neg p \vee \neg q)$

b- $(\neg p \vee \neg q)$

c- $(p \wedge \neg q)$

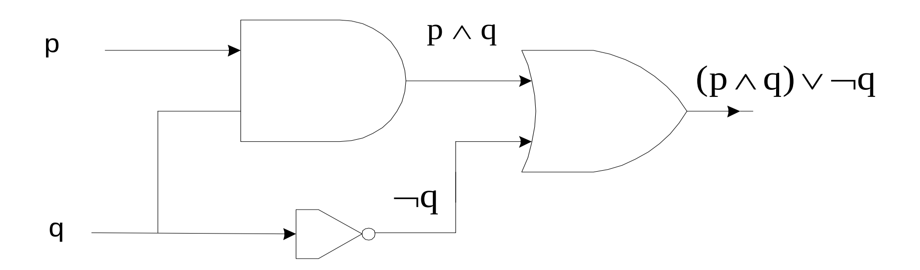
d- $p \wedge (q \wedge r)$

e- $p \vee (q \wedge r)$

f- $\neg(q \vee r)$

g- $p \wedge (q \vee \neg p)$

4-Find the out put of A, B, and C, for all possible of the following circuit.



5- بين مخطط الدوائر المنطقية للجمل التالية واوجد المخرجات النهائية لكل الامكانيات المدخلة لها:

a- $(p \vee q) \wedge (\neg p \vee \neg q)$

b- $(\neg p \vee \neg q)$

c- $(p \wedge \neg q)$

d- $p \wedge (q \wedge r)$

e- $p \vee (q \wedge r)$

f- $\neg(q \vee r)$

g- $p \wedge (q \vee \neg p)$

6- **Show how to construct a circuit to test for the truth of** ($p \oplus q$)

الفصل السادس

الخوارزميات

Algorithms

1-6 المقدمة

الخوارزمية (***Algorithm***)هي طريقة لحل بعض المسائل خطوة بخطوة , وهي كذلك مجموعة محددة من الاوامر او التعليمات (*instructions*) لها الخصائص التالية:

1- الدقة ***precision*** : الخطوات توضع او ينصُّ عنها بدقة لكي يمكن كتابة الخوارزمية في لغات البرمجة وتنفيذها بواسطة الحاسوب.

2- خاصية الحصول على نتيجة واحدة بكل خطوة بعد تنفيذها ***uniqueness***: حيث النتائج الوسطية (*intermediate results*) لكل خطوة بعد تنفيذها تعرًف بنتيجة وحيدة , اي كل خطوة وسطية للخوارزمية تنتج نتيجة واحدة فقط وتعتمد تلك النتيجة على المدخلات وعلى نتائج الخطوات التي تسبقها فقط .

3- خاصية المحدود ***Finiteness*** : وهي ان الخوارزمية تتوقف بعد عدة اوامر محددة قد تم تنفيذها.

4- المدخلات ***input*** : الخوارزمية تستلم مدخلات اي ندخل لها قيم

5- المخرجات ***output***:الخوارزمية تنتج مخرجات اي نحصل على النتيجة المطلوبة من الخوارزمية

6- العمومية ***generality*** : الخوارزمية تطبَّق الى مجموعة من المدخلات ولم تخص بيانات دون اخرى

ان الخصائص المذكورة اعلاه يمكن توضيحها من خلال المثال التالي:

مثال 6-1:

اعتبر الخوارزمية التالية هي لايجاد الرقم الاكبر من بين ثلاثة ارقام a ,b و c

1. $x := a.$
2. if $b > x$, then $x := b.$
3. if $c > x$, then $x := c.$

ان فكرة الخوارزمية هنا هي للتفتيش عن الارقام واحد واحد ونسخ الاكبر في المتغير x ونتيجة الخوارزمية ان x ستكون مساوية الى الرقم الاكبر من بين الارقام الثلاثة.

ان الرمز $(y := z)$ تعني انسخ القيمة z في y او ابدل القيمة الحالية لـ y بقيمة z وعندما تنفَّذ $(y := z)$ فان قيمة z لن تتغير , ونسمي $(:=)$ اداة التعيين(*assignment operator*). والان لنرى كيف ان الخوارزمية اعلاه تنفَّذ لبعض القيم المحددة لـ a ,b و c .ان مثل هذه المحاكاة (*simulation*) تدعى بأقتفاء الاثر **trace** . اولاً نفرض:

$$a = 1, \qquad b = 5, \qquad c = 3$$

في السطر الاول اعلاه وضعنا x تساوي a(1) وفي السطر الثاني: $(1 > 6)$ $b > x$ وهي جملة خاطئة . اذن نضع x مساوية الى b(5) , وفي السطر الثالث $(3 > 5)$ $c > x$, وهي جملة خاطئة , لذلك لانعمل شيء. وعند تلك النقطة فان قيمة x هي 5 وتعتبر القيمة الاكبر بين b, a و c .

ولو فرضنا ان:

$$a = 6, \qquad b = 1, \qquad c = 9$$

في السطر الاول اعلاه وضعنا x تساوي a(6) وفي السطر الثاني: $(1 > 6)$ $b > x$ وهي جملة خاطئة .لذلك لانعمل شيء, وفي السطر الثالث $(9 > 6)$ $c > x$, وهي جملة صحيحة . اذن نضع x مساوية الى 9 . وعند تلك النقطة فان قيمة x هي 9 وتعتبر القيمة الاكبر بين a ,b و c .

وكما اشرنا في بداية هذا الموضوع بان خطوات الخوارزمية يجب ان تنص بشكل واضح , فان في مثالنا هذا ان الخطوات وضعت بدقة وبوضوح لكي الخوارزمية يمكن كتابتها في لغات البرمجة ومن ثم تنفيذها بواسطة الحاسوب.

من جانب اخر باعطاء قيم المدخلات فان كل خطوة وسطية للخوارزمية تنتج نتيجة واحدة . على سبيل المثال اعطاء القيم التالية:

$$a = 1, \qquad b = 5, \qquad c = 3$$

ففي السطر 2 فان x اعطيت القيمة 5 بغض النظر عما يقوم بالتنفيذ للخوارزمية شخص او حاسوب .

ان الخوارزمية لاحظناها تتوقف بعد عدد من الخطوات المحددة التي تجيب السؤال المعطى. فعلى سبيل المثال ان الخوارزمية المذكورة توقفت بعد ثلاثة خطوات وانتجت الرقم الاكبر من بين القيم الثلاث المعطاة للاعداد a ,b و c . ايضاً لاحظنا ان الخوارزمية استلمت مدخلات

وهي القيم الثلاث المعطاة للاعداد a ,b و c . واعطت المخرجات المتمثلة بقيمة x . واخيرا تبين لنا الخوارزمية انها عامة اي يمكننا ان نجد العدد الاكبر لاي ثلاثة اعداد بغض النظر عن قيمهم . وبذلك حققت الخوارزمية المذكورة في هذا المثال جميع الخواص التي ذكرناها في مدخل هذا الموضوع.

مثال 6-2: اكتب خوارزمية لايجاد العدد الاولي الخمسين ثم اطبعه ؟

Problem: Find and print the 50th prime number

**A prime number is a whole number not evenly divisible by any other number other than 1 and itself*

ان المسالة المراد حلها هنا بشكل خوارزمية , هي ايجاد العدد الاولي الخمسين وطباعته , حيث ان العدد الاولي هو العدد بكامله والذي لايقسم باي عدد (زوجي او فردي) اخر غير نفسه والعدد 1 (اي لايقبل القسمة الاّ على نفسه وعلى واحد فقط).

اما الخوارزمية هنا فهي كما في الخطوات التالية:

1- انشئ قائمة بكل الاعداد الاولية N_1, N_2, N_3 ....

2- رتِّب القائمة المذكورة تصاعدياً.

3- اطبع العنصر رقم خمسين في القائمة .

4- انهي الخوارزمية.

1. Generate a list of all prime numbers $N_1, N_2, N_3 \ldots$
2. Sort the list into ascending order
3. Print out the 50th element in this list
4. end

مثال 6-3

مسألة: اذا اعطيت عددين صحيحين موجبين , احسب قاسمهما المشترك الاعظم

لحساب ذلك نستخدم الخوارزمية الاقليدية **Euclid's algorithm التالية:**

1- الخطوة 1: احصل على قيم عددين صحيحين من المستخدم user .

2- الخطوة 2: عيِّن M الى قيمة العدد الاكبر و N الى قيمة العدد الاصغر.

3- الخطوة 3: اقسم M بواسطة N وسمي الباقي بـ R.

4- الخطوة 4 : اذا كان R ليس صفراً , من ثم عيِّن قيمة N الى M وعيِّن قيمة R الى N وارجع الى الخطوة 2 , وعكس ذلك فان القاسم المشترك الاعظم هو القيمة التي تم تعيينها تواً الى N.

- Problem: Given two positive integers, compute their greatest common divisor
- **Euclid's algorithm:**

  Step 1: Get two positive integer values from the user

  Step 2: Assign M and N the value of the larger and smaller of the two input values, respectively

  Step 3: Divide M by N, and call the remainder R

  Step 4: If R is not 0, then assign M the value of N, assign N the value of R, and return to step 2; otherwise, the greatest common divisor is the value currently assigned to N

2-6 مجموعة الرموز للخوارزميات Notation for algorithms

بالرغم من ان اللغة الاعتيادية (*ordinary language*) مناسبة لوصف الخوارزمية , فان عدد من علماء الرياضيات والحاسوب يفضلون لغة الكود الزائفة (*pseudocode*) بسبب دقتها وتركيبها وشموليتها . وسميت (*pseudocode*) بهذا الاسم بسبب انها تشبه الكود الحقيقية (*resemble actual code*) او البرامج (*programs*) للغات البرمجة مثل باسكال وس++ (Pascal and C++) . وهناك عدة اشكال من لغة الكود الكاذبة او الزائفة . وبعكس لغات البرمجة الحقيقية التي تستخدم الفاصلة والحروف الكبيرة والصغيرة وبعض الكلمات الخاصة فان اي شكل من الكود الكاذبة يكون مقبولاً مادامت الاوامر فية غير مبهمة وتشبه شكل الكود . وكمثال على ذلك نعيد كتابة الخوارزمية التي ذكرناها سابقا الخاصة بايجاد العدد الاكبر من بين ثلاثة اعداد .

Algorithm Of Finding The Maximum Of Three Numbers 3-6 خوارزمية ايجاد الاكبر لثلاثة اعداد

هذه الخوارزمية تجد الاكبر للاعداد a ,b و c

1- المخلات: Input : ثلاثة اعداد a ,b و c

2- المخرجات output : هي قيمة *x* , التي تمثل العدد الاكبر للاعداد a ,b و c

```
1.  procedure max(a,b,c)
2.  x:=a
3.  if b>x  then      // if b larger than x, update x
4.  x:=b
5.  if c>x  then      // if c larger than x, update x
6.  x:=c
7.  return(x)
8.  end max
```

لاحظ ان خوارزميتنا تشتمل على عنوان (ايجاد الاكبرللاعداد لثلاثة اعداد) وصف مختصر للخوارزمية , المخلات والمخرجات والخطوات التي يتم اجراءها (*procedures*) وهي هنا اجراء واحد. ولجعل الاجراءات ملائمة فاننا نشير الى اسطر الاجراءات بارقام . فالاجراء في الخوارزمية اعلاه يشمل اسطر مرقمة . ان السطر الاول يحتوي على كلمة اجراء (*procedure*), اسم الاجراء وهو الدالة *max* , ومابين قوسي الدالة توجد العوامل *parameters* التي تستخدم في الاجراء .

ان العوامل تصف البيانات (***data***) , المتغيرات (***variables***) , المصفوفات (***arrays***) التي تتوفر للاجراء . وفي الخوارزمية اعلاه فان العوامل تمثل ثلاثة اعداد هي a ,b و c , وفي نهاية الاجراء يوجد سطر يحوي كلمة **end** متبوعة باسم الاجراء اي ان هذا السطر يعني نهاية الاوامر التي يتضمنها الاجراء . ومن هنا تكون اوامر الاجراء التي يتم تنفيذها محصورة بين اسطر كلمة (*procedure*) وكلمة **end** , وهي الاسطر من 2 الى 7 .

وعنما تنفذ الخوارزمية اعلاه بدءاً من السطر **2** فاننا نضع ***x*** مساوية الى ***a*** , وفي السطر **3** تتم مقارنة قيمة **b** مع ***x*** . فاذا كانت قيمة b اكبر من x فاننا ننفذ السطر 4 , اي

$$x := b$$

لكن اذا كانت *b* ليست اكبر من *x* فاننا نقفز الى السطر 5 , وفي هذا السطر فان *c* و *x* تتم مقارنتهما . فاذا كانت *c* اكبر من *x* فان السطر **6** يتم تنفيذه:

$$x := c$$

لكن اذا كانت **c** ليست اكبر من **x** فاننا نقفز الى السطر 7 , وفي هذا السطر فان **x** تمثل القيمة الصحيحة للعدد الاكبر من بين الاعداد a ,b و c . وفي السطر 7 فاننا نرجع قيمة *x* التي تعني العدد الاكبر الى الدالة ***max*** او ماتسمى بـ ***invoker*** , وهنا تنتهي الاجراء (*procedure*) ان الخوارزمية اعلاة قد اوجدت بشكل صحيح العدد الاكبر لثلاثة اعداد , ويمكننا كتابة برنامج بلغة C++ لايجاد العدد الاكبر كمايلي:

```
// Finding the maximum of three integers                 ايجاد العدد الاكبر لثلاثة اعداد
#include <iostream.h>
int maximum( int , int , int );// function prototype
 int main()                                          الدالة الرئيسية
 {
int a, b, c;                                         تعريف الاعداد الثلاثة

cout << "Enter three integers: ";
cin >> a >> b >> c;                                  ادخال الاعداد الثلاثة
  // a, b and c below are arguments to  the maximum function call      استدعاء الدالة
cout << "Maximum is: " << maximum( a, b, c ) << endl;
return 0;
  }
  // Function maximum definition ,where  x, y and z below are parameters to
 // the maximum function definition
 int maximum( int x, int y, int z )                  تعريف الدالة
  {
  int max = x;
if ( y > max )                                       اجراء المقارنات
max = y;
if ( z > max )
 max = z;
 return max;}
```

برنامج يبين تعريف دالة ايجاد العدد الاكبر لثلاثة اعداد باستخدام لغة C++

وبشكل عام ان تركيب $if\ -then$: **if – then structure** هو كمايلي:

$$\begin{array}{lll} if & x & then \\ & action1 & (statement1) \\ & action2 & (statement2) \end{array}$$

فاذا كان الشرط x صحيحاً (*true*) , فان الـ ***action*** تنفذ والتحكم يمر الى الجملة التي تلي الـ ***action*** . اما اذا كان الشرط خاطئ ***false*** , فان التحكم ينتقل مباشرة الى الجملة التي تلي الـ ***action*** .

هناك صيغة مختلفة للجملة الشرطية هي $if\ -then-else$ وتركيبها هو:

$$\begin{array}{lll} if & x & then \\ & action1 & statement1 \\ else & & \\ & action2 & statement2 \end{array}$$

وهذا يعني اذا كان الشرط x صحيحاً ***true*** فان الـ ***action1*** ينفَّذ فقط دون ***action2*** ومن ثم التحكم ينتقل لتنفيذ الجملة بعد الـ ***action2*** , اما اذا كان الشرط خاطئ فان الـ ***action2*** ينفَّذ فقط دون ***action1*** ومن ثم ينتقل التحكم لتنفيذ الجملة بعد الـ ***action2*** . وكما نلاحظ من صيغ الجمل الشرطية اعلاه فاننا نترك فراغ لتعريف الجمل التي تكوّن الـ ***actions*** (يعني جمل ***action*** زاحفة قليلا باتجاه اليمين عند الكتابة عما هو للبقية).

من جانب اخر اذا كان الـ ***action*** في جملة ***if-then*** يتضمن عدة جمل , فاننا نحصر تلك الجمل بالكلمات **begin** و **end** . انظر المثال التالي:

$$\begin{array}{ll} if\ \ x \geq 0\ \ then & \\ \quad begin & \\ \quad x := x-1 & statement1 \\ \quad a := b+c & statement2 \\ \quad end & \end{array}$$

من ناحية اخرى اذا وجدنا خطين مائلين (// ***two slash***)كما هو في خوارزمية ايجاد الاكبر انفة الذكر , فانهما يشيران الى بداية الملاحظة (*comment*),التي تمتد الى نهاية السطر بعد (//) والتي لاتدخل في اوامر التنفيذ بل انها تساعد القارئ على فهم الخوارزمية .

ان جملة ***return(x)*** تنهي الاجراء وترجع قيمة ***x*** الى الدالة ***max*** , كما ان جملة ***return*** بدون **x** تنهي ببساطة الاجراء ايضاً. اما اذا لم توجد جملة ***return*** فان الاجراء ينتهي فقط قبل السطر **end** . ان الاجراء الذي يحوي جملة ***return(x)*** هو دالة ومجالها هو كل القيم النافذة للعوامل والمدى (*range*) هو مجموعة كل القيم التي ترجَّع بواسطة الاجراء . وعندما نستخدم الكود الكاذبة(***pseudocode***) فاننا سنستخدم الادوات الرياضية (***mathematics operators***) مثل (+, - , *, و/) بالاضافة الى الادوات العلائقية (***relational operators***) مثل $(=,\neq,\leq,\geq,<,>)$, وكذلك الادوات المنطقية (***logical operators***) مثل (AND,OR, NOT) , وسنستخدم الاداة (=) للدلالة على المساواة ونستخدم (=:) للدلالة على اداة تعيين القيمة ***Operator Assignment***) .

ان اسطر الاجراء (*procedure*) التي تنفذ بالتتابع (*sequentially*) هي عادة ماتكون جمل التعيين (*assignment*) , جمل شرطية , جمل تكرارية (*loops*) , جمل ارجاع (*return statements*) او من جمل خليط من كل ماذكر من تلك الجمل .

ان احدى تراكيب التكرار (*loop structures*) المفيدة هي : **while loop** وتركيبها كما يلي:

$$\textit{while}\ \ x\ \ \textit{do}$$
$$\textit{action}1 \qquad \textit{statement}1$$

التي فيها الـ (*action*) يعاد تنفيذه مادامت *x* هي صحيحة ***true*** . ونسمي الـ (*action*) هيكل التكرار *body loop* وكما في جملة ***if*** الشرطية , فانه اذا احتوى الـ (*action*) على عدة جمل فاننا ايضاً نستخدم الكلمات **begin** و **end** . وسنوضح جملة (*while loop*) في المثال القادم الذي يجد فيه القيمة الاكبر في المتتالية (*sequence*) وسنستخدم ترقيم الاسطر المكونة للخوارزمية ونحدِّث المتغيرات التي تمثل القيمة الاكبر .

مثال 6-4:

لاي ثلاثة اعداد يتم ادخالها عن طريق المستخدم احسب مجموعها ومتوسطها واظهر النتيجة .

Problem: For any three numbers input by the user, compute their sum and average and output them

ان مدخلات الخوارزمية اعلاه هي المتغيرات c,b,a , واما المخرجات فهي المجموع sum والمتوسط الحسابي average وهما ايضا متغيرات . ولكتابة الخوارزمية نتبع الخطوات التالية:

1- الخطوة الاولى : احصل على قيم المتغيرات c,b,a من المستخدم .

2- الخطوة الثانية :اجعل المتوسط للمتغيرات الثلاث هو avrg مساوياً الى $avrg = (a+b+c)/3$

3- الخطوة الثالثة :اجعل المجموع sum للمتغيرات الثلاث هو $sum = (a+b+c)$

4- الخطوة الرابعة : اطبع المجموع والمتوسط

1-Get the values of a, b, c from user

2-Set avrg to $(a+b+c)/3$

3-Set sum to $(a+b+c)$

4-Print sum and avrg

4-6 خوارزمية ايجاد العنصر الاكبر في متتالية محددة باستخدام جملة while loop

Algorithms of finding the largest element in a finite sequence

في هذه الخوارزمية نجد العدد الاكبر في المتتالية $s_1, s_2, ..., s_n$ وتستخدم نسخة الخوارزمية هذه جملة التحكم **while loop** .

المدخلات : **input** هي المتتالية $s_1, s_2, ..., s_n$ وطول المتتالية n

المخرجات **Output** وهي العنصر الاكبر في المتتالية وهو هنا *large*

الكود الكاذبة (*pseudocode*) هي كمايلي:[1]

```
1.  procedure find-large(s , n)
2.  large := s_1
3.  i:=2
4.  while i ≤ n    do
5.  begin
6.  if  s_i > large    then   // a larger value was found القيمة الاكبر قد وجدت
7.  large := s_i
8.  i := i + 1
9.  end
10. return(large)
11. end find-large
```

نتتبع الخوارزمية هذه عندما $(n = 4)$ و s هي المتتالية:

$$s_1 = -2, \quad s_2 = 6, \quad s_3 = 5, \quad s_4 = 6$$

في السطر الثاني وضعنا ***large*** مساوية الى قيمة s_1 , وفي هذه الحالة $large = -2$ () , وفي الخطوة اللاحقة وضعنا قيمة i مساوية الى 2 . في السطر 4 اختبرنا قيمة i فيما لو كانت مساوية او اقل من n ($i \le n$) . في هذه الحالة اختبرنا فيما لو كانت $2 \le 4$. وبما ان هذا الشرط هو صحيح true , فاننا ننفذ هيكل جملة التكرار *while* من الاسطر 5-9 . ففي السطر 6 اختبرنا فيما لو كانت s_i اكبر من *large* ($s_i > large$) , وفي هذه الحالة اختبرنا فيما لو كانت s_2 اكبر من *large* ($s_2 > large$) اي $6 > -2$. وبما ان هذا الشرط هو صحيح true فاننا ننفذ السطر 7 , ومن خلال ذلك يكون *large* قد وضعت قيمته مساوية الى 6 ($large = 6$) . وفي السطر 8 فان i قد وضعت قيمتها 3 . ثم رجعنا الى السطر 4.

ومن جديد اختبرنا قيمة i فيما لو كانت مساوية او اقل من n ($i \le n$) . في هذه الحالة اختبرنا فيما لو كانت $3 \le 4$. وبما ان هذا الشرط هو صحيح true , فاننا ننفذ هيكل جملة التكرار **while** . وفي السطر 6 اختبرنا فيما لو كانت s_i اكبر من *large* ($s_i > large$) , وفي هذه الحالة اختبرنا فيما لو كانت s_3 اكبر من *large* ($s_3 > large$) اي $5 > 6$. وبما ان هذا الشرط هو ***false*** , فاننا لا ننفذ السطر 7 , بل نذهب الى السطر 8 . وفي السطر 8 فان i قد وضعت قيمتها 4 . ثم رجعنا مرة اخرى الى السطر4.

ومن جديد اختبرنا قيمة i فيما لو كانت مساوية او اقل من n ($i \le n$) . في هذه الحالة اختبرنا فيما لو كانت $4 \le 4$. وبما ان هذا الشرط هو صحيح ***true*** , فاننا ننفذ هيكل جملة التكرار **while** . وفي السطر 6 اختبرنا فيما لو كانت s_i اكبر من *large* ($s_i > large$) , وفي هذه الحالة اختبرنا فيما لو كانت s_4 اكبر من *large* ($s_4 > large$) اي $6 > 6$. وبما ان هذا الشرط هو ***false*** , فاننا لا ننفذ السطر 7 , بل نذهب الى السطر 8 . وفي السطر 8 فان i قد وضعت قيمتها 5 . ثم رجعنا مرة اخرى الى السطر4.

ومن جديد اختبرنا قيمة i فيما لو كانت مساوية او اقل من n ($i \le n$) . في هذه الحالة اختبرنا فيما لو كانت $5 \le 4$. وبما ان هذا الشرط هو خاطئ ***false*** , فاننا ننهي جملة التكرار **while** ونصل الى السطر **10** , حيث نرجع قيمة *large* (6) وهنا قد وجدنا العنصر الاكبر في المتتالية . ويلاحظ اننا استخدمنا المتغير i من خلال قيمه من 1 الى n لفحص عناصر المتتالية وايجاد العنصر الاكبر. وهناك جملة تحكم تكرارية اخرى اكثر استخداماً من جملة التحكم **while** والتي تستخدم بدلها.وهي جملة **for** وصيغتها هي:

```
for var := init to limit do

action
```

وكما في جملة **if** وجملة **while** السابقتين , اذا كان الـ ***action*** يشتمل على عدة جمل فاننا نستخدم كلمتا **begin** و **end** لحصر تلك الجمل . وعندما تنفذ جملة for , فان الـ ***action*** ينفّذ لقيم المتغير من القيمة الابتدائية ***init*** الى القيمة النهائية ***limit***.وبدقة اكثر ان ***init*** و ***limit*** هما تعابير رياضية لها قيم اعداد صحيحة , حيث المتغير ***var*** اولاً توضع له القيمة الابتدائية ***init*** . اذا كان ***var*** اصغر او يساوي ***limit*** $(var \le limit)$, فاننا ننفّذ الـ ***action*** ثم نزيد **1** الى المتغير ***var*** . وهنا العملية تتكرر . ان التكرار يستمر حتى نصل الى ان ***var*** اكبر من ***limit*** . $(var > limit)$, اي ان الـ ***action*** لاينفّذ على الاطلاق.

6-5 خوارزمية ايجاد العنصر الاكبر في متتالية محددة باستخدام جملة for

Algorithms of finding the largest element in a finite sequence

في هذه الخوارزمية سنجد العنصر الاكبر في المتتالية **s** ولكن باستخدام جملة **for** :

المدخلات : **input** هي المتتالية $s_1, s_2 ......, s_n$ وطول المتتالية **n**

المخرجات **output** وهي العنصر الاكبر في المتتالية وهو هنا ***large***

الكود الكاذبة (*pseudocode*) هي كمايلي:

1. **procedure** *find-large*(*s* ,*n*)
2. $large := s_1$
3. **for** i:=2 to n **do**
4. if $s_i > large$ $then$ // a larger value was found القيمة الاكبر قد وجدت
5. $large := s_i$
6. ***return***(*large*)
7. ***end*** *find-large*

ويمكن تحويل الخوارزمية اعلاه الى **برنامج بلغة C++** كمايلي:

```
#include<iostream.h>

int larg(int,int,int,int); // the procedure; prototype function        نموذج اولي لتعريف الدالة

int large;
int s[4]={2,14,60,10}; the array of elements        عناصر المتتالية
int main()
{
cout<<larg(s[1],s[2],s[3],s[4]); // call function        استدعاء الدالة وعرض العدد الاكبر
cout<<endl;
return 0;
}
int larg(int a,int b,int c,int d) // function definition        تعريف الدالة
{
large = a;
for (int i=2;i< 4;i++)        // for  جملة التحكم
if (s[i] >large )
large = s[i];
return (large);        ارجاع العدد الاكبر الى الاستدعاء اعلاه
}
```

عندما نطوِّر الخوارزمية . فان الفكرة الجيدة غالباً ما تكون هي تقسيم المسألة الاصلية الى مسألتين او اكثر .

والاجراء (*procedure*) بالامكانتطويره لحل كل جزء من المسألة الاصلية ومن ثم بعد ذلك تجمع الحلول للحصول على حل للمسألة الاصلية وكما سنرى في المثال القادم: افترض اننا نريد خوارزمية لايجاد العدد الاولي الاصغر(*least prime number*) الذي يزيد على العدد الصحيح الموجب المعطى, بدقة اكثر , المسألة هي :

اذا اعطيت عدد صحيح موجب ، فاوجد العدد الاولي الاصغر p الذي يحقق العلاقة $(p > n)$.

ولحل تلك المسألة فاننا نقسّمها الى مسألتين . الاولى نطوّر خوارزمية لتحدّد فيما لو كان العدد الصحيح الموجب عدداً اولياً ام لا. ومن ثم نستخدم تلك الخوارزمية لايجاد العدد الاولي الاصغر والذي هو اكبر من العدد الصحيح الموجب المعطى في المسألة .

ففي الخوارزمية الاتية سنختبر فيما لو كان العدد الصحيح الموجب m هو عدد اولي ام لا. ببساطة نختبر فيما لو كان اي عدد صحيح بين 2 و ***m-1*** (حيث 1 ليس عدداً اولياً) يقبل القسمة على **m**. فاذا وجدنا عدداً بين ***m-1*** , 2 يقبل القسمة على ***m*** , فان ***m*** ليس عدداً اولياً. واذا فشلنا في تحقيق ذلك فان العدد **m** هو عدد اولي . ان الخوارزمية الاتية تبين اننا نسمح للاجراءات (*procedures*) لارجاع ***true*** او ***false*** في الاختبار.

6-6 خوارزمية اختبار العدد الصحيح الموجب فيما لو كان عدداً اولياً ام لا

Algorithm of testing whether a positive integer is prime

هذه الخوارزمية تختبر العدد الصحيح الموجب ***m*** فيما لو كان عدداً اولياً ام لا. والنتيجة *output*: هي ***true*** اذا كان **m** عدداً اولياً. و *false* اذا لم يكن ***m*** عدداً اولياً.

المدخلات: ***input*** : ان **m** عدد صحيح موجب.

المخرجات **output** : هي ***true*** اذا كان ***m*** عدداً اولياً. و ***false*** اذا لم يكن ***m*** عدداً اولياً.

```
procedure is_prime(m)
for i:=2 to m-1 do
if m  mod  i = 0 then          // i divides m ,          i تقسّم m
return (false)
return(true)
end is_prime
```

اما الخوارزمية الاتية في لايجاد العدد الاولي الاصغر الذي يزيد العدد الصحيح الموجب **n** مستخدمين الخوارزمية السابقة (الجزء الاول لحل المسألة). ولغرض استدعاء الاجراء الذي يرجِّع قيمة كما هو في الخوارزمية السابقة , فاننا نسميها. فلاستدعاء عملية الاجراء (*procedure*) المسماة ***proc*** والتي لاتعيد قيمة , فاننا نكتب :

$$call \quad proc(\, p_1 , p_2 ,.......p_k \,)$$

حيث ان $(\, p_1 , p_2 ,.......p_k \,)$ تمثل العوامل (*parameters*) التي تمرَّر الى ***proc*** .

7-6 خوارزمية ايجاد عدد اولي اكبر من عدد صحيح معطى

algorithm of finding a prime larger than a given integer

هذه الخوارزمية تجد اصغر عدد اولي الذي يزيد على العدد الصحيح الموجب ***n***

المدخلات: ***input*** : ان ***n*** عدد صحيح موجب

المخرجات ***output*** ان **m** هي العدد الاولي الاصغر واكبر من **n**

```
procedure large-prime(n)
m := n+1
while not is-prime (m) do
m := m+1
return(m)
end large-prime
```

وبما ان عدد الاعداد الاولية غير محدد فان الاجراء (*procedure*) في هذه الخوارزمية ستنتهي اخيراً *eventually* (*terminate*).

8-6 الخوارزمية الاقليدية : The Euclidean Algorithm

وهي اقدم واشهر خوارزمية لايجاد القاسم المشترك الاعظم لعددين صحيحين(*greatest common divisor of two integers* , حيث ان القاسم المشترك الاعظم لعددين *m* و *n* (قيمهما ليس صفراً) هو اكبر عدد صحيح موجب يقسِّم كل من *m* و *n* . على سبيل المثال ان القاسم المشترك الاعظم للعددين 4 و 6 هو 2 (اي كلاهما يقبل القسمة على **2**) , والقاسم المشترك الاعظم للعددين **3** و **8** هو **1** .

اننا نستخدم فكرة (*notion*) القاسم المشترك الاعظم وذلك عندما نفحص لنرى فيما لو كان الكسر (m/n) في الحدود الدنيا *lowest) terms* اي قيمته **1** , ويعني ذلك انه اذا كان القاسم المشترك الاعظم لـ m و n هو **1** , فاننا نقول ان الكسر (m/n) في الحدود الدنيا *lowest terms* , وعكس ذلك فاننا نقلل قيمة الكسر . وعلى هذا الاساس فان $(4/6)$ ليس في الحدود الدنيا لان القاسم المشترك الاعظم لهما هو **2** وليس **1**. بينما الكسر $(3/8)$ هو في الحدود الدنيا لان القاسم المشترك لهما هو **1** كما قلنا . وبعد هذه المقدمة عن قابلية الاعداد للقسمة فاننا نختبر القاسم المشترك الاعظم بالتفصيل ونقدم الخوارزمية الاقليدية كما يلي[1]:

اذا كانت ***a*** , ***b*** و ***q*** اعداداً صحيحة , $(b \neq 0)$, تخضع للعلاقة التالية:

$$(a = bq)$$

فاننا نقول ان ***b*** تقسِّم ***a*** ونكتبها بـ $(b \mid a)$. وفي هذه الحالة فاننا نسمي q بناتج القسمة (*quotient*) ونسمي b بالمقام او المقسوم عليه (*divisor*) لـ ***a*** . اما اذا كانت ***b*** لاتقِّسم ***a*** فاننا نكتبها بـ $(b \mid a)$.

مثال 6-5:

بما ان $(21 = 3.7)$, فان 3 تقسِّم 21 , ونكتبها $(3 \mid 21)$, وناتج القسمة (quotient) هو 7.

فاذا فرضنا ان m و n هما عددان صحيحان وليس كلاهما صفراً . فان من بين كل الاعداد الصحيحة التي تقسِّم m و n , هنك عدد اكبر مقسوم علية يعرف بالقاسم المشترك الاعظم للعددين m و n .ويكتب بالصيغة التالية:

$$gcd(m,n)$$

ان القواسم الموجبة (*positive divisors*) للعدد **30** هي:

1, 2, **3**, **5**, 6, 10,**15**, 30

و **105** هي:

1, **3**, **5**, 7, **15**, 21, 35, 105

بينما القواسم المشتركة الموجبة لـ **30** و **105** (تمثل الارقام الغامقة في العددين المذكورين) هي:

1, 3, 5, 15

ونستنتج من ذلك ان القاسم المشترك الاعظم لـ **30** و **105** هو:

$$gcd(30,105) = 15$$

نظرية 1-6 Theorem

افرض ان m , n و c اعداد صحيحة فانه:

1- اذا كانت c قاسم مشترك a common divisor لـ m و n فان :

$$c\,|\,(m+n)$$

2- - اذا كانت c قاسم مشترك a common divisor لـ m و n فان :

$$c\,|\,(m-n)$$

3- اذا كانت $(c\,|\,m)$, فان $(c\,|\,mn)$

من جانب اخر اذا كانت ***a*** عدد صحيح (*nonnegative integer*) وكانت ***b*** عدد صحيح موجب , فان :

$$(a \quad mod \quad b)$$

هو باقي قسمة ***a*** على ***b*** . فعلى سبيل المثال: $(105 \quad mod \quad 30 = 15)$. **ان الخوارزمية الاقليدية اسست على الحقيقة التالية:**

اذا كان $(r = a \quad mod \quad b)$, فان :

$$gcd(a,b) = gcd(b,r)$$

وقبل برهنة ذلك نوضِّح كيف ان الخوارزمية الاقليدية تُستخدم لايجاد القاسم المشترك الاعظم .

مثال 6-6: بما ان :

$$(105 \quad mod \quad 30 = 15)$$

فانه بواسطة الحقيقة اعلاه ان:

$$gcd(105,30) = gcd(30,15)$$

وبما ان :

$$(30 \mod 15 = 0)$$

اذن مرة ثانية بواسطة الحقيقة اعلاه ان :

$$gcd(30,15) = gcd(15,0)$$

وبواسطة الفحص او التمحيص *inspection* , $(gcd(15,0) = 15)$, فاننا على اساس ذلك نجد ان :

$$gcd(105,30) = gcd(30,15) = gcd(15,0) = 15$$

ومن ناحية اخرى اذا كانت ***a*** عدد صحيح (*nonnegative integer*) وكانت ***b*** عدد صحيح موجب , و :

$$r = (a \mod b)$$

فان :

$$gcd(a,b) = gcd(b,r)$$

والان نكتب الخوارزمية الاقليدية كخوارزمية كمايلي[1] :

المدخلات ***inputs*** : ***a*** و ***b*** , اعداد غير سالبة وليست كلاهما صفراً

.(*a and b are nonnegative integers , not both zero*

المخرجات***output*** : القاسم المشترك الاعظم لـ ***a*** و ***b*** .

1. **procedure** $gcd(a,b)$
2. // make a largest
3. **if** $a < b$ **then**
4. *swap*(*a*,*b*)

   // *that is execute*

   // *temp*:=*a*

   // *a*:=*b*

   // *b*:=*temp*

5. **while** $b \neq a$ **do**

6. **begin**

7. *r:= a mod b*

8. *a:=b*

9. *b:=r*

10. **end**

11. **return**(a)

12. **end** *gcd*

والخوارزمية اعلاه يمكننا تحويلها الى برنامج بلغة C++ انظر المثال (6-8) .

مثال 6-7 : اوجد القاسم المشترك الاعظم $gcd(504,396)$

الحل:

افرض ان $a = 504$ و $b = 396$. بما ان $a > b$, اذن نتحرك للخطوة **5** في الخوارزمية الاقليدية اعلاه. وبما ان $b \neq 0$, فاننا نتقدم الى الخطوة 7 ونضع (*r:= a mod b*) , اي:

$$a \bmod b = 504 \bmod 396 = 108$$

ومن ثم نتحرك الى الخطوة **8** و **9** :

a:=b

b:=r

اي $b = 108$ $a = 396$ ونرجع الى الخطوة 5 . وبما ان $b \neq 0$ فاننا نتقدم الى الخطوة 7 .ونضع

r:= a mod b

$a \bmod b = 396 \bmod 108 = 72$. ومن ثم نتحرك الى الخطوة **8** و **9** :

a:=b

b:=r

اي $b = 72$ $a = 108$ ونرجع الى الخطوة **5** . وبما ان $b \neq 0$ فاننا نتقدم الى الخطوة **7**. ونضع

r:= a mod b

. $a \bmod b = 108 \bmod 72 = 36$

ومن ثم نتحرك الى الخطوة **8 و 9** :

a:=b

b:=r

اي $b = 36$ $a = 72$, ونرجع الى الخطوة **5** . وبما ان $b \neq 0$ فاننا نتقدم الى الخطوة **7** . ونضع

r:= a mod b

. $a \bmod b = 72 \bmod 36 = 0$

ومن ثم نتحرك الى الخطوة **8** و **9** :

a:=b

b:=r

اي $b = 0$ $a = 36$, ونرجع الى الخطوة **5** . وبما ان $b = 0$ فاننا نتقدم الى الخطوة **11** , ونرجع $a(36)$ وهو مايمثل القاسم المشترك للعددين 396 و 504 .

اننا نلاحظ من تنفيذ الخوارزمية اعلاه ان جملة التحكم ***while*** (الخطوات من 5- 11) دائما تنتهي عند الخطوات **8** و **9** ومن ثم قيم **a** و **b** تتحدث (*update*) الى القيم الاصغر . وبما ان العدد الصحيح غير السالب لايمكن تقليليه وبالتحديد ***b*** اصبح صفراً وجملة التحكم تتوقف نهائياً. وبواسطة الخوارزمية المذكورة فان القيمة المرجعة في الخطوة 11 هي $gcd(a,b)$

9-6 الخوارزميات التكرارية الذاتية Recursive Algorithms

ان الاجراء التكراري (*recursive procedure*) هو الاجراء الذي يستدعي نفسه عدة مرات . اما الخوارزمية التكرارية فانها الخوارزمية التي تشتمل على الاجراء التكراري. ان التكرار الذاتي (*recursion*) هو اداة قوية واسلوب طبيعي لحل صنف كبير من المسائل . فالمسألة في هذا الصنف يمكن حلها وذلك بواسطة تكنيك يسمى جزِّء وفرِّق تسد (*divide-and-conquer*) الذي فيه المسألة تجزء الى مسائل من نفس النوع لما هو للمسألة الاصلية , وبالمقابل كل مسألة جزئية تقسَّم الى اخرى مسائل جزئية وتستمر العملية الى ان نصل الى مسألة يتم حلها بطريقة بسيطة (*straightforward way*) . واخيرا تجمع حلول المسائل الجزئية لنحصل على الحل الخاص بالمسألة الاصلية .

تذكر انه اذا كانت n اكبر او تساوي 1 $(n \geq 1)$, فان مضروب n $(n!)$ هو:

$$n! = n.(n-1).(n-2).....2.1$$

واذا كانت $(n \geq 2)$ فاننا ايضا نكتبها بنفس الطريقة السابقة اي :

$$n! = n.(n-1).(n-2).....2.1$$

واذا تفحصنا الصيغة اعلاه وتركنا n جانباً فان المتبقي من الصيغة هو $(n-1)!$, اي ان الصيغة السابقة يمكن كتابتها :

$$n! = n.(n-1)!$$

والتي يجب ان تكون صحيحة حتى لو كانت $n = 1$

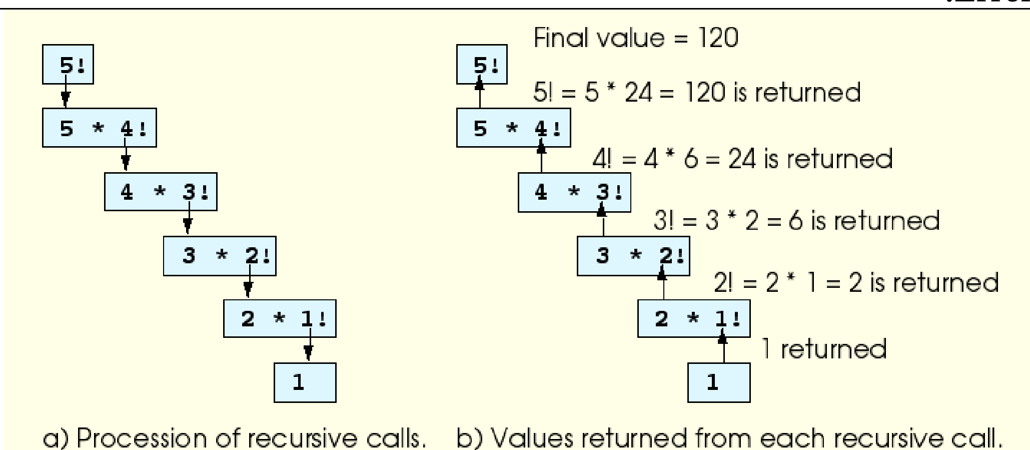
فعلى سبيل المثال: $5!$ تكتب بـ

$$5! = 5.4.3.2.1 = 5.4!$$

ان مسألة حساب $5!$ تقلل لحساب $4!$, وحساب $4!$ يقلل لحساب $3!$, و $3!$ تقلل لحساب $2!$ و $2!$, تقلل لحساب $1!$, و $1!$ تقلل لحساب $0!$, و $0!$ لايقلل بل من التعريف يساوي 1 .

وبعد تجزئة المسألة ($5!$) الى مسائل اصغر , يبدا الان وبشكل معكوس حل المسائل الاصغر ابتداءاً من الاصغر وهي الصفرهنا ووضع نتيجة الحل في الخطوة التي تسبقها اي $1!$, وهذه في التي تسبقها اي $2!$, وهكذا الى ان نصل الى نتيجة $5!$ وهي 120 ,اي حاصل ضرب ($4!$ x 5 =120).انظر الشكل (6-1) التالي [8] :

!Error



الشكل (6-1) يبين حساب مضروب !5 بطريقة التكرار الذاتي العكسية *recursion*

والان نكتب الخوارزمية لحساب المضروب *factorial* هي كمايلي:

10-6 خوارزمية حساب *n*! : Algorithm Of Computing *n* Factorial

المدخلات *input* : *n* عدد صحيح اكبر او يساوي صفراً

المخرجات *output* : هي مضروب *n* ($n!$)

1.	**procedure** *factorial* (*n*)
2.	**if** *n=0* **then**
3.	**return** (1)
4.	**return**(*n** *factorial*(*n-1*))
5.	**end** *factorial*

ويمكننا كتابة برنامجاً بلغة السي++ لحساب مضروب *n* ($n!$) , **حيث n من 0 الى 10** , بطريقة التكرار الذاتي (***recursion***) **كمايلي:**

```
// Recursive factorial function                    دالة المضروب التكرارية
#include  <iostream.h>
#include  <iomanip.h>

unsigned long  factorial(  unsigned long  );        التعريف الاولي لدالة المضروب

int  main()
{
for  (  int  i = 0; i <= 10; i++ )                  الاعداد من 0 الى 10
cout << setw( 2 ) << i << "! = " << factorial( i) << endl;
return  0;
}

// Recursive definition of function factorial        تعريف الدالة

unsigned long  factorial(  unsigned long  number )
{
if  ( number <= 1 )   // base case                   القاعدة الاساسية
return  1;
else                    // recursive case             حالة التكرار الذاتي
return  number * factorial( number - 1 );
}
```

ومن اجل المقارنة بين الطريقة التكرارية الذاتيـة اعـلاه و الطريقـة غـير التكراريـة ***non recursively*** التاليـة مسـتخدمين جملة التحكم ***for*** يكون البرنامج كمايلي:

```
factorial =1;  بداية التعريف

for ( int counter = number; counter >= 1; counter-- )      تكرار تناقصي اعتيادي

  factorial *= counter;    حساب المضروب
```

OR

```
unsigned long fact (unsigned long n)
 {
    int i;
    unsigned long f=1;
    for (i=0; i<=n; i++)
        f*=i;
    return f;
}
```

وفي حالة القاسم المشترك الاعظم الانف الذكر اذا اردنا ايجاده باستخدام تكنيك التكرار الذاتي recursionفاننا نعتمد النقطتين المهمتين التاليتين:

1- ان القاسم المشترك الاعظم لاي عدد والصفر هو العدد نفسه

2- ان القاسم المشترك الاعظم للرقم الاول (الاكبر) والرقم الثاني (الاصغر) هو نفس القاسم المشترك الاعظم للرقم الثاني والرقم المتبقي من حاصل قسمة الرقم الاول على الرقم الثاني.

***Recursive Algorithm*:**

1. GCD of any number and 0 is the number itself.

2. GCD of the first number (bigger) and the second number (smaller) is the same as GCD of the second number and the remainder of the division of the first number on the second number.

وسنوضح تلك النقطتين في المثال التالي:

مثال 6-8: اكتب دالة التكرار الذاتي (*recursion function*) لايجاد القاسم المشترك الاعظم **GCD** لاي عددين صحيحين . اكتب البرنامج بلغة السي++

Write a recursive function to find GCD of two integers.

```
#include<iostream.h>
int gcd(int,int);
int main()
{int x,y;
cout<<"Enter tow integar"<<"\n";
cin>>x>>y;
cout<<gcd(x,y)<<"\n";
return 0;  }
int gcd (int a, int b){
if (b==0)
return a;
else
return gcd ( b,a%b);}
```

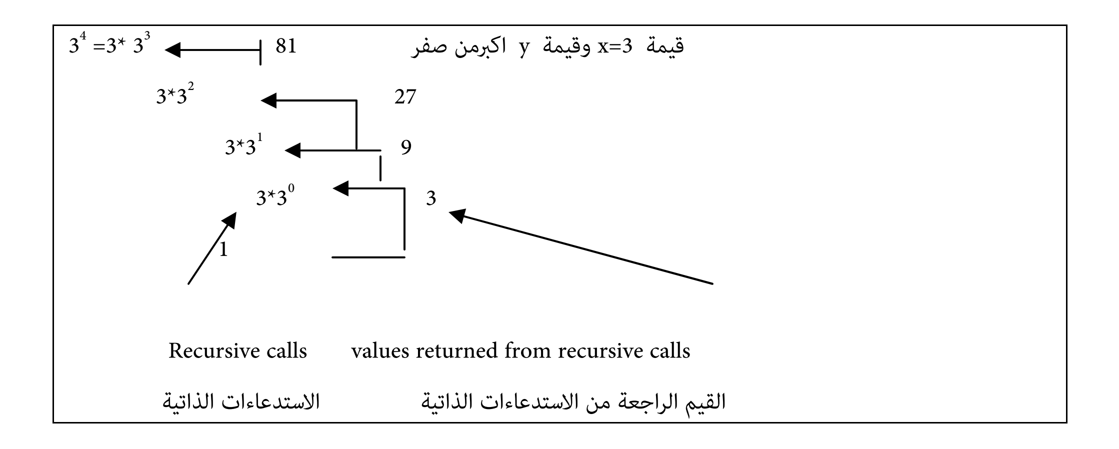
ولحساب قيمة دالة الأُس (*function power*) بطريقة الـ (***recursion***) فاننا نعتمد مايلي:

$$x^y = \begin{cases} 1 & \text{if } y = 0 \\ x * x^{y-1} & \text{if } y > 0 \end{cases}$$

مثال 6-9:

احسب قيمة 3^4 باستخدام طريقة التكرار الذاتي؟

الحل:



مثال 6-10 :

اكتب دالة التكرار الذاتي (*recursion function*) اللازمة لحساب قيمة x^y, استخدم لغة C++ .

```
#include<iostream.h>
unsigned long  power(int x, int y);
int main()
{
int x,y;
cout<<"Enter x , y values"<<"\n";
cin>> x>>y;
cout<<"power(x,y)= "<<power(x,y)<<endl;
return 0; }
unsigned long  power(int x, int y)  {
      if (y==0)
          return 1;
      else
         return x*power(x, y-1);  }
```

11-6 مقدار الزمن وسعة الخزن للخوارزميات (تعقيد الخوارزمية)

Complexity Of Algorithms

بالرغم من ان برنامج الحاسوب يشتق من خوارزمية صحيحة , لكنه قد يكون غير مفيـد لانـواع معينـة مـن القـيم المدخلـة للبرنامج بسبب ان الزمن الذي نحتاجة لتنفيذ البرنامج او لخزن البيانات , المتغيرات (*variables* , الخ .. كبير جداً.

ان تحليل الخوارزمية(*analysis of an algorithm*) يشير الى عملية تقدير الزمن وسعة الخزن اللازمـة لتنفيـذ البرنـامج. ان مايسمى بـ (*complexity of an algorithm*) هو الاشارة الى مقدار الزمن وسعة الخزن المطلوبة لتنفيذ الخوارزمية . وهنـا في هذا الجزء من الكتاب سنركز على مسالة تقدير الزمن المطلوب لتنفيذ الخوارزميات.تصوًر اننا اعطينا المجموعـة *X* لـــ ***n*** من العناصر ,حيث قسم منها بعلامة حمراء والقسم الاخر بعلامة سوداء . واردنا ان نجد المجموعة الجزئية لـ *X* *subset of* (*X*) التي تحوي على الاقل عنصر واحد احمر .

افترض اننا انشأنا خوارزمية التي تختبر (*examine*) كل المجموعات الجزئية لـ *X* وتعد التي تحوي على الاقل عنصرـ احمـر واحد ومن ثم نستخدم تلك الخوارزمية كبرنامج حاسوب. فبما ان المجموعة التي لها ***n*** من العنـاصر يكـون لهـا 2^n مـن المجموعات الجزئية (راجع فصل المجموعات) , فان البرنامج يحتاج على الاقل 2^n من وحدات الزمن للتنفيذ.

وبغض النظر عما هي وحدات الزمن , فان 2^n تتزايد (تكبر) بسرعة كلـما ازدادت ***n*** (انظـر الجـدول(6-1) ادنـاه [1]), ذلك ماعدا القيم الصغيرة لـ ***n*** التي يجب ان تكون معقولة (*infeasible*) لتنفيذ البرنامج .

Number of steps to termination for input of size n	*Time to execute if n =* ,*n* زمن التنفيذ مقابل قيم			
	3	**6**	**9**	**12**
1	10^{-6} sec	10^{-6} sec	10^{-6} sec	10^{-6} sec
lg lgn	10^{-6} sec	10^{-6} sec	2×10^{-6} sec	2×10^{-6} sec
lg n	2×10^{-6} sec	3×10^{-6} sec	3×10^{-6} sec	4×10^{-6} sec
n	3×10^{-6} sec	6×10^{-6} sec	9×10^{-6} sec	10^{-5} sec
n lg n	5×10^{-6} sec	2×10^{-5} sec	3×10^{-5} sec	4×10^{-5} sec
n^2	9×10^{-6} sec	4×10^{-5} sec	8×10^{-5} sec	10^{-4} sec
n^3	3×10^{-5} sec	2×10^{-4} sec	7×10^{-4} sec	2×10^{-3} sec
2^n	8×10^{-6} sec	6×10^{-5} sec	4×10^{-4} sec	4×10^{-3} sec
*(lg n denotes $\log_2 n$: the logarithm of n to base 2)				

جدول (6-1) يمثل زمن تنفيذ خوارزمية اذا كان زمن تنفيذ الخطوة الواحدة هو 1 مايكروثانية وكذلك ين عدد خطوات انهاء البرنامج للمدخلات من حجم n مقارنة بزمن التنفيذ[1].

ويصل زمن التنفيذ الى (4×10^{16} yr) عندما تكون n=100 وعندما تكون عدد الخطوات لانهاء البرنامج هي 2^n مثلاً.

ان تحديد عوامل الكفاءة (*performance parameters*) لبرنامج الحاسوب هي امر صعب ويعتمد على عدد العوامل التي سبق وان تعامل الحاسوب بمثلها و على طريقة تمثيل البيانات وكيف ان الحاسوب يترجمها الى لغة الالة *machine language*). وبالرغم من ان الدقة في تقدير الزمن اللازم لتنفيذ البرنامج يجب ان تاخذ في الاعتبار جملة من العوامل (*factors*) فان المعلومات المفيدة (*useful information*) بالامكان ان يتم الحصول عليها بواسطة تحليل زمن الخوارزمية الاساسية.

ان الزمن الذي نحتاجه لتنفيذ الخوارزمية هو دالة للقيم المدخلة input , وعادة لايمكن الحصول على صيغة مبسطة لتلك الدالة ويقع اختيارنا على الاقل.وبدلاً من ان نتعامل مع المدخلات مباشرة فاننا نستخدم عوامل التي تخص حجم تلك المدخلات . نحن يمكننا ان نطلب الزمن الاقل الذي نحتاجه لتنفيذ الخوارزمية من بين كل المدخلات ذات الحجم ***n*** . وهذا الزمن يسمى زمن احسن حالة (*the best case-time*) للمدخلات من الحجم ***n*** , ويمكننا ايضا ان نطلب اكبر زمن نحتاجه لتنفيذ الخوارزمية بين كل المدخلات من الحجم ***n*** . وهذا الزمن يسمى زمن الحالة الاردأ (*worst –case time*) للمدخلات من الحجم ***n*** . وحالة مهمة اخرى هي زمن الحالة المعدل (*average-case time*) وهي معدل الزمن الازم بتنفيذ الخوارزمية على بعض المجموعات المحددة من المدخلات كلها من الحجم ***n*** .

يمكننا ان نقيس الزمن الذي نحتاجه بواسطة الخوارزمية وذلك بواسطة عد الاوامر المنفذة . وبشكل بديل بالامكان استخدام الزمن المقدّر على سبيل المثال حساب عدد المرات لكل تكرار (*loop*) يتم تنفيذه . واذا كان مبدأ الخوارزمية يعتمد على المقارنات (*comparisons*) كما يحصل في برامج الترتيب (*Sort programs*), فاننا نعدُّ ايضاً عدد المقارنات .

مثال 6-11:

ان التعريف المعقول لحجم المدخلات لخوارزمية ايجاد القيمة الاكبر في متتالية محددة هو عدد العناصر في المتتالية المدخلة وان التعريف المعقول لزمن التنفيذ هو عدد تكرار جملة التحكم while loop (راجع الخوارزمية المذكورة). بتلك التعريفات فان اردأ واحسن ومتوسط الزمن لتلك الخوارزمية لمدخلات من الحجم n هي (n-1) لكل منهم بسبب ان جملة التحكم while loop دائما يتم تنفيذها (n-1) من المرات.[1]

من جانب اخر اننا لسنا مهتمين بالضبط باحسن واردأ زمن نحتاجه للخوارزمية لُتنفَّذ مقارنة بانه كيف ان احسن وراسء زمن ينمو (يكبر) grow كلما ازداد حجم المدخلات . على سبيل المثال:افترض ان اسوء زمن the worst case time لخوارزمية هو :

$$t(n) = 60n^2 + 5n + 1$$

وكان حجم القيم المدخلة n . فمن المعادلة المذكورة نرى انه عندما n كبيرة , فان الحد $60n^2$ فانه تقريباً يكون مساوياً الى $t(n)$ (انظر الجدول(6-2) ادناه). وبذلك فان $t(n)$ تكبر مثل $60n^2$.

n	$t(n) = 60n^2 + 5n + 1$	$60n^2$
10	6051	6000
100	600,501	600,000
1000	60,005,001	60,000,000
10000	60,000,050,001	6,000,000,000

جدول (6-2) ين مقارنة التزايد لـ $t(n)$ مع $60n^2$

فاذا كانت المعادلة اعلاه تمثل الزمن بالثواني فان المعادلة التالية هي لحساب الزمن بالدقائق لمدخلات بحجم n :

$$T(n) = n^2 + \frac{5}{60}n + \frac{1}{60}$$

ان هذا التغيير بوحدة الزمن لايؤثر على كيفية تزايد او نمو الزمن كلما ازدادت المدخلات, لكنه فقط يمثل وحدات الزمن التي بها نقيس حالة اسوء زمن لمدخلات بحجم n . وعلى اساس ذلك اننا نصف كيف ان اسوأ واحسن زمن يكبر كلما ازدادت المدخلات , فاننا لسنا نبحث عن الحد الغلب (*dominant term*) (على سبيل المثال هنا $60n^2$) بل ايضا نهمل المعاملات الثابتة (*constant coefficients*). وتحت تلك الفرضيات فان $t(n)$ تكبر مثل n^2 (بعد حذف الثابت 60) كلما n تزداد. و نقول ان $t(n)$ هي بمرتبة n^2:

(*f*(*n*) *is an order of g*(*n*)) ونكتبها :

$$t(n) = \Theta(n^2)$$

وتقرأ ان $t(n)$ هي ثيتا (Θ) لـ n^2 : n^2 is theta of $t(n)$

ان الفكرة الاساسية هي لاستبدال تعبير مثل : $t(n) = 60n^2 + 5n + 1$ بتعبير ابسط مثل n^2 الذي يكبر(يتزايد) بنفس نسبة كبر $t(n)$. وفيمايلي تعريفات الصيغ الاخرى[1] :

تعريف 6-2:

افترض ان f و g هما دوال بمجال *domain* هو مجموعة الاعداد {1,2,3.....} , فاننا نكتب :

$$f(n) = O(g(n))$$

ونقول ان $f(n)$ بمرتبة على الاغلب $g(n)$: (*(f(n) is an order at most g(n)*) اذا كان يوجد ثابت موجب C_1 مثل ذلك:

$$|f(n)| \leq C_1 |g(n)|$$

لكل الاعداد الصحيحة ولكن الاعداد الصحيحة الموجبة بالتحديد.

ونكتب:

$$f(n) = \Omega(g(n))$$

ونقول ان $f(n)$ بمرتبة على الاقل $g(n)$: (*(f(n) is an order at least g(n)*) اذا كان يوجد ثابت موجب C_2 مثل ذلك:

$$|f(n)| \geq C_2 |g(n)|$$

لكل الاعداد الصحيحة ولكن الاعداد الصحيحة الموجبة بالتحديد.

واخيراً نكتب :

$$f(n) = \Theta(g(n))$$

ونقول ان :

f(n) is an order of g(n) **if** $f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$

ان تعبير مثل $f(n) = O(g(n))$ يشير احياناً الى رمز مايعرف **Big Oh Notation** لـ f , وبشكل مشابه فان $f(n) = \Omega(g(n))$ يشير الى مايسمى بـ ***Omega Notation*** لـ f , واخيراً فان $f(n) = \Theta(g(n))$ يشير الى مايسمى ***Theta Notation*** .

مثال 6-12 :

بما ان :

$$t(n) = 60n^2 + 5n + 1$$

فان :

$$60n^2 + 5n + 1 \le 60n^2 + 5n^2 + n^2 = 66n^2 \qquad for \quad n \ge 1$$

وسنأخذ هنا قيمة C_1 مساوية الى 66 ($(C_1 = 66)$ وكما اشرنا في التعريف السابق , فاننا نحصل على :

$$,60n^2 + 5n + 1 = O(n^2)$$

وبما ان :

$$60n^2 + 5n + 1 \ge 60n^2 \qquad for \quad n \ge 1$$

فاننا ناخذ $C_2 = 60$ وكما اشرنا بالتعريف ايضاً فاننا نحصل على :

$$,\ 60n^2 + 5n + 1 = \Omega(n^2)$$

وبما ان :

$60n^2 + 5n + 1 = O(n^2)$ و $60n^2 + 5n + 1 = \Omega(n^2)$ فان :

$$60n^2 + 5n + 1 = \Theta(n^2)$$

مثال 6-13:

افرض ان:

$$a_k n^k + a_{k-1} n^{k-1} + ... + a_1 n + a_0$$

هي متعدد الحدود *polynomial* في *n* لـ *k* درجة (n^k) , حيث a_i غير سالبة *nonnegative* , فاثبت ان:

$$a_k n^k + a_{k-1} n^{k-1} + ... + a_1 n + a_0 = \Theta(n^k)$$

البرهان:

افرض ان :

$$C_1 = a_k + a_{k-1} + ... + a_1 + a_0$$

فان:

$$a_k n^k + a_{k-1} n^{k-1} + ... + a_1 n + a_0 \le a_k n^k + a_{k-1} n^{k-1} + ... + a_1 n^k + a_0 n^k$$
$$= (a_k + a_{k-1} + ... + a_1 + a_0) n^k = C n^k$$

وعلى اساس ذلك فان:

$$a_k n^k + a_{k-1} n^{k-1} + ... + a_1 n + a_0 = O(n^k)$$

وبما ان:

$$a_k n^k + a_{k-1} n^{k-1} + ... + a_1 n + a_0 \ge a_n n^k$$
$$a_k n^k + a_{k-1} n^{k-1} + ... + a_1 n + a_0 = \Omega(n^k)$$

اذن:

$$a_k n^k + a_{k-1} n^{k-1} + ... + a_1 n + a_0 = \Theta(n^k)$$

وهو المطلوب .

مثال 6-14:

ان $lg\ n$ يشير الى لوغاريتم n للاساس 2, وبما ان :

$$lg\ n < n \qquad for \quad n \ge 1$$

فان:

$$2n + 3\,lg\ n < 2n + 3n = 5n \qquad for \quad n \ge 1$$

لذلك فان:

$$2n + 3\,lg\ n = O(n)$$

وكذلك بما ان:

$$2n + 3\,lg\ n \ge 2n \qquad for \quad n \ge 1$$

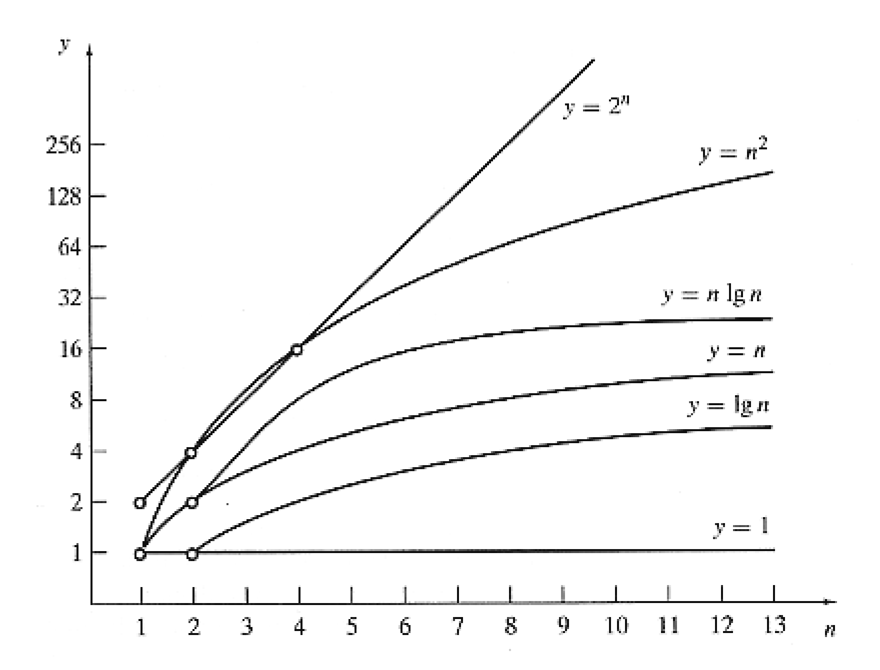
فان:

$$2n + 3\,lg\ n = \Omega(n)$$

وعلى اساس ذلك فان:

$$2n + 3\lg n = \Theta(n)$$

وفـــــيما يــــلي شـــــكل (6-2) يوضــــح تزايـــــد (*growth*) بعـــــض الـــــدوال الشـــــائعة (*common function*):[1]



شكل (6-2) يوضح تزايد بعض الدوال الشائعة

تمارين الفصل السادس

1- اكتب خوارزمية لايجاد العدد الاكبر من بين 5 أعداد ثم نفِّذ تلك الخوارزمية.

2- اكتب خوارزمية لايجاد العدد الاكبر والعدد الاصغر في مصفوفة

3- اكتب خوارزمية لمعرفة فيما لو كان العدد زوجياً ام فردياً.

4- اوجد القاسم المشترك الاعظم gcd(108,72) مستخدماً الخوارزمية الاقليدية.

5- اكتب خوارزمية لايجاد x^y بطريقة التكرار الذاتية.

6- اكتب خوارزمية لترتيب عناصر مصفوفة مكونة من 100 عنصر.

7- بيِّن كيف ان الخوارزمية في الجزء 3-6 تجد العدد الاكبر للاعداد

$$a = 5, \qquad b = -3, \qquad and \qquad c = 6$$

8- اكتب خوارزمية تجمع اعداد المتتالية $s_1, s_2, \ldots.., s_n$.

9- اكتب خوارزمية لأيجاد العنصر الاصغر والاكبر في المتتالية $s_1, s_2, \ldots.., s_n$

الفصل السابع

مبادئ نظرية المخططات

Graphs Theory Concepts

7-1 المقدمة:

ان المخططات (Graphs) هي صور (*Pictures*) نستخدمها لبيان علاقة ما بشكل مرئي *to visualized a relation*), وهي تختلف عما نعنيه بالرسم البياني لعلاقة ما كالدوال (*functions*) في علم الجبر . ان المخططات ترسم عادة بنقاط Dots وخطوط (Lines) , وبما انها ترسم كذلك , فانها تبدو كخرائط الطرق (Road Maps) . ان النقاط في المخطط تدعى (Vertices) ويرمز لها بـ v_1 , v_2........ , والخطوط التي تربط بين تلك النقاط تدعى بالحدود او الاضلاع (Edges) , ويرمز لها بـ e_1 , e_2......... .واذا بدأنا الحركة مثلاً من النقطة v_0 الى v_1 عبر الحد (Edge) الذي يربطهما ومن ثم الى نقطة اخرى حتى نصل الى النقطة v_n مثلاً فان الجولة الكاملة بين تلك النقاط نطلق عليها بالمسار (**Path**). ان الموقع الحقيقي لتلك النقاط وطول وشكل الحدود لايخبرنا شيئاً عن العلاقة , ولهذا السبب ومن المفيد ان نضع تعريفاً للمخطط الذي لايذكر النقاط الهندسية (*geometric points*) او الاسهم التي تربط بينها . ان مانعنيه هو ان نعرّف المخطط بدلالة المجموعات , وفي هذا الفصل تم الاعتماد على المصدر الاول في تحليل المواضيع التالية , ونبدأها بالتعريف التالي:

تعريف (7-1):

Graph

Informally, a graph is a finite set of dots called **vertices** (or **nodes**) connected by links called **edges** (or **arcs**). More formally: a **simple graph** is a (usually finite) set of vertices V and set of unordered pairs of distinct elements of V called edges.

Not all graphs are simple. Sometimes a pair of vertices are connected by multiple edge yielding a multigraph At times vertices are even connected to themselves by a edge called a loop, yielding a pseudographd. Finally, edges can also be given a direction yielding a directed graph (or **digraph**).

ان المخطط (Graph) (او **المخطط غير المباشر** ***undirected Graph***) , **G** , يشتمل على مجموعة (**Set**) تـدعى V , عناصرها تدعى النقاط (*vertices or nodes*) , وعلى مجموعة من الحدود او الاضـلاع (*Edge or arcs*) تـدعى E , ومِثـل ذلك فان اي حد (*edge*) $(e \in E)$ يرتبط مع **زوج غير مُرتَّب** مـن النقـاط (*Unordered pair of vertices*). فمـثلاً اذا كان هناك حد وحيد (*Unique edge*) (e) يضم النقاط ***v*** و ***w*** , فاننا نكتب :

$$e = (w, v) \quad \text{او} \quad e = (v, w)$$

وفي هذه الحالة فان (v, w) ترمز الى الحد (*edge*) بين v و w في مخطط غير مباشر وفي زوج (v, w) غير مرتب كما هو واضح اعلاه لقيم (*e*) .

اما المخطط المباشر (***a Directed graph***) , **G** , او مايسمى مختصراً (**Digraph**) , فانه يشتمل على مجموعـة (Set) من النقاط (*vertices or nodes*) تدعى **V** وعلى مجموعة من الحدود او الاضلاع (*Edge or arcs*) تدعى E , مثـل ذلـك فان اي حد (*edge*) $(e \in E)$ يرتبط مع **زوج مرتب** من النقاط (*ordered pair of* vertices).

A digraph (or a **directed graph**) is a graph in which the edges are directed. (Formally: a digraph is a (usually finite) set of vertices V and set of *ordered* pairs (*a*,*b*) (where *a*, *b* are in V) called edges. The vertex *a* is the **initial vertex** of the edge and *b* the **terminal vertex**.

فمثلاً اذا كان هناك حد وحيد(*Unique edge*) (e) يضم النقاط v و w في زوج مرتب (v, w) من النقاط , فاننا نكتـب :

$$e = (v, w)$$

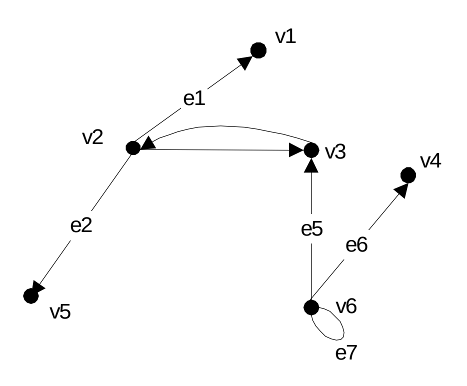
التي ترمز الى الحد من v الى w .من جانب اخر ان الحد (*edge*) في مخطط سواء كان مباشراً ام غير مبـاشر الـذي يكـون مرتبطاً مع زوج من النقاط v و w يقال عنه بانه يشكل جزءاً طبيعياً (*incident on*) من v و w , وان v و w يقال انهما واقعتان على الحد (*e*) ويكونان نقاط متجاورة (*adjacent vertices*). واذا كان **G** هو مخطـط مبـاشر او غير مباشر مكوّن من نقاط (*vertices*) (V) ومن حدود (*Edges*) (E) , فاننا نكتبه كمايلي:

$$G = (V, E)$$

حيث V, E يفترض انهما هما مجموعتـان محـدودتان (**Finite Sets**) وان V يفـترض ان تكـون مجموعـة غـير خاليـة (**Nonempty set**).

مثال 7-1:

ان الشكل (7-1) يشير الى مخطط موجه مباشر (*directed graph*) ,(والحافات المتجهة المباشرة (*directed edges*) ممثلة بالاسهم . ان الضلع او الحد (e_1) يرتبط بالزوج المُرتّب (v_2, v_1) من النقاط (*vertices*) , والضلع او الحد (e_7) يرتبط مع الزوج المُرتّب (v_6, v_6) من النقاط . ان الحد (e_1) يرمز له بـ (v_2, v_1) و الحد (e_7) يرمز له بـ (v_6, v_6) , وهكذا لبقية النقاط والاضلاع المبينة في هذا الشكل.



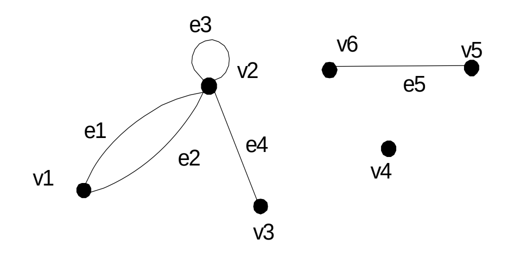
شكل (7-1) يوضّح مخطط موجه مباشر (*directed graph*)

ان التعريف (7-1) يسمح بحدود منفصلة (*distinct edges*) لتكون مرتبطة بنفس ازوج من النقاط (*vertices*), على سبيل المثال في الشكل (7-2) الحدود (e_7) و (e_2) كلاهما مرتبط بزوج النقاط (*vertex* pair) $\{v_1, v_2\}$, ان مثل تلك الحدود تدعى بالحدود المتوازية (*parallel edges*).

ان الحد الذي يقع على نقطة واحدة يدعى بـ (**loop**) او الحلقة .

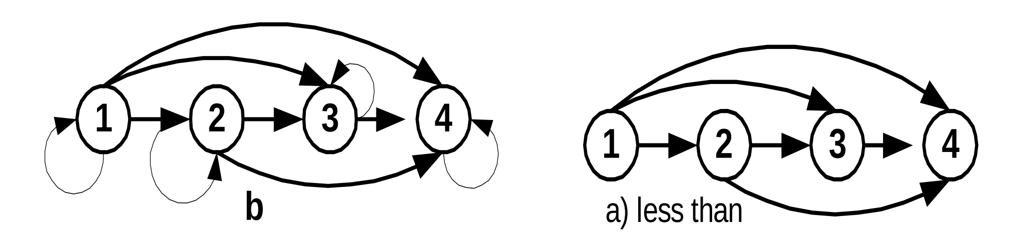
A loop is an edge that connects a vertex to itself

على سبيل المثال: في الشكل (7-2) ان الحد $e_3 = (v_2, v_2)$ هو يشكل مايسمى حلقة او (**loop**) . ان النقطة مثل (v_4) لاتقع على اي حد ولذا فانها تسمى نقطة معزولة او مفصولة (*an isolated vertex*) . اخيراً ان المخطط (**Graph**) الذي لايحوي (**loop**) او حدود متوازية يدعى المخطط البسيط (**Simple Graph**) .



شكل (7-2) يبين الحدود المتوازية والـ loop

ونود ان نذكر هنا ايضاً انه يمكن رسم المخطط المتجه (*Directed Graph*) من خلال رسم دوائر صغيرة بدلاً من النقاط وتسمى ايضاً (*vertices*) وذلك لكل عنصر من عناصر المجال (*Domain*) والمدى (*range*) للعلاقة (*relation*) التي نريد ان نمثلها بالمخطط. فالشكل (7-3) ادناه يمثل مخطط العلاقة اقل من او اقل او تساوي الى العلاقة على المجموعة {4,3,2,1}.



شكل (7-3) يمثل مخطط العلاقة **اقل من** او **اقل او تساوي** الى العلاقة على المجموعة {4,3,2,1}.

7-2 المخططات المتشابهة او المتماثلة Similarity Graphs

لفهم مخططات التماثل (**Similarity Graphs**) نعطي المثال التالي:

هذا المثال متعلق بمسألة تقسيم الاشياء الى اصناف بشكل مجموعة اعتماداً على خصائصها. على سبيل المثال, افترض ان خوارزمية خاصة مكتوبة بلغة ++C من قبل عدد من الاشخاص واردنا ان نقسّم برامج الاشخاص الى اصناف معتمدين على حواص معينه لتلك البرامج كالتي مدونة في الجدول (7-1) ادناه [1].

program	Number of program lines	Number of return statements	Number of function calls
1	60	20	1
2	41	10	2
3	68	5	8
4	90	34	5
5	75	12	14

جدول(1-7) ين برامج ++C التي تعالج نفس الخوارزمية

افترض اننا اعتمدنا على الخواص التالية في التقسيم:

1- عدد الاسطر في البرنامج (**Number of Lines In a Program**)

2- عدد جمل **Return** في البرنامج

3- عدد استدعاءات الدالة في البرنامج (**Function Calls**)

ان المخطط المتشابه او المتماثل (Similarity Graph) في هذا المثال بني كمايلي:

ان النقاط (Vertices) تطابق البرامج. والنقطة يرمز لها بـ (p_3, p_2, p_1), حيث (p_i) هي قيمة الخاصية i اي البرنامج رقم i , وعرفنا دالة عدم التشابه (**Dissimilarity Function**) s كمايلي:

لأي زوج من النقاط ان :

$$w = (q_1, q_2, q_3) \text{ و } v = (p_1, p_2, p_3)$$

واستخدمنا العلاقة التالية:

$$s(v, w) = |p_1 - q_1| + |p_2 - q_2| + |p_3 - q_3|$$

فاذا فرضنا ان (v_i) تمثل البرنامج i . فانه من خلال تطبيق المعادلة اعلاه نحصل على مايلي:

$$s(v_1, v_2) = |66 - 41| + |20 - 10| + |1 - 2| = 36$$

$$s(v_1, v_2) = 36$$

وبنفس الطريقة نجد العلاقات التالية:

$$s(v_1, v_3) = 24$$

$$s(v_1, v_4) = 42$$

$$s(v_1, v_5) = 30$$

$$s(v_2, v_3) = 38$$

$$s(v_2, v_4) = 76$$

$$s(v_2, v_5) = 48$$

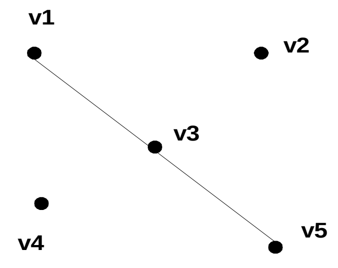
$$s(v_3, v_4) = 54$$

$$s(v_3, v_5) = 20$$

$$s(v_4, v_5) = 46$$

فـاذا كانـت (v) و (w) هـما نقـاط طبقـاً الى برنـامجين , فـان $s(v\ ,w)$ تقـيس كـم عـدم التشـابه للبرنـامجين (*Dissimilar The Programs*). ان القيمة الكبيرة لـ $s(v\ ,w)$ تعني عدم التشـابه كبـير والعكـس ان القيمـة الصـغيرة تبين التماثل او التشابه بين البرنامجين . وباستخدام قيمة عددية ثابتة S , فاننا نضع حافة (Edge) بين (v) و (w) اذا كانت قيمة $s(v\ ,w)$ اقل من S (وبشكل عام انه هناك مخططات متشابهة مختلفة لقـيم مختلفـة مـن S) . كـما اننـا نقول ان (v) و (w) هما في نفس الصنف (**Class**) اذا كانت $(v\ = w)$ او هناك مسار من (v) الى (w) .

ان مخطط التشابه (**Similarity Graph**) المبين بالشكل (7-4) ادناه هو طبقاً الى البرامج في الجدول (7-1) مع تثبيـت عدد يساوي $S = 25$. في هذا المخطط ان البرامج صنفت الى ثلاثة اصـناف كمايـلي: $\{1,3,5\}$, $\{2\}$ و $\{4\}$, علـماً بان لأية مسالة حقيقية يمكننا اختيار قيمة S بالتجريب لاغراض الدقة .



شكل (7-4) يبين مخطط التشابه للبرامج المبينة في الجدول (7-1) مع قيمة S=25

3-7 المسارات والدورات الكاملة Path and cycles

اذا تصورنا ان النقاط في المخطط تمثل المدن والحدود (*edges*) تمثل الطرق , فان المسار هو طبقاً الى رحلة تبدأ من مدينة يمر من خلال عدة مدن وينتهي بمدينة ما. ونعبر عن المسار بشكل صيغة كما في التعريف التالي:

تعريف 7-2:

افرض ان v_0 و v_n هما نقاط في المخطط Graph . فمسار من v_0 الى v_n بطول n هو متسلسلة متغيرة (*alternating sequence*) من $n+1$ من النقاط و n من الحدود (*edges*) تبدأ بالنقطة v_0 وتنتهي بالنقطة v_n.

A path is a sequence of consecutive edges in a graph and the length of the path is the number of edges traversed

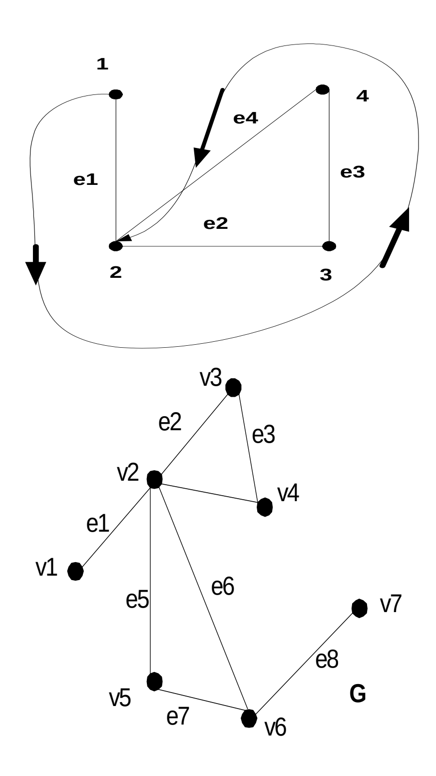
$$(v_0, e_1, v_1, e_2, v_2, \ldots.., v_{n-1}, e_n, v_n)$$

وفيها الحد e_1 يقع بين النقطة v_{n-i} و v_i حيث $(i = 1,2\ldots\ldots..n)$. ان الصيغة المبينة في التعريف تعني نبدأ عند النقطة v_0 والذهاب على طول الحد e_1 الى النقطة v_1 ومن ثم الذهاب على طول e_2 الى v_2 وهكذا.

مثال 7-2:

في الشكل (7-5) ادناه ان المسار: $(1, e_1, 2, e_2, 3, e_3, 4, e_4, 2)$

هو من النقطة v_1 الى النقطة v_2 وبطول 4 . والمسار (6) بطول 0 هو من النقطة v_6 الى نفس النقطة v_6



شكل (7-5) اليسار يبن المسارات $(1, e_1, 2, e_2, 3, e_3, 4, e_4, 2)$ بطول 4 والمسار (6) بطول 0

واليمين : يبين طول المسار 4 وذلك من النقطة 1 الى 2 , ومن 2 الى 3 , 3 الى 4 , ومن 4 الى 2.

وبغياب الحدود المتوازية في الاشارة الى المخطط فاننا نطمس الحدود . علـى سـبيل المثـال: المسـار(1.2.6) ربمـا يكتـب ايضـاً كمايلي:

$$(1,2,3,4,2)$$

ان المخطط المتصل **(Connected Graph)** هو المخطط الذي فيه يمكننا الوصول من نقطة الى اخرى على المسار . والصيغة التي تعبر عن ذلك هي كما في التعريف ادناه:

تعريف 7-3 :

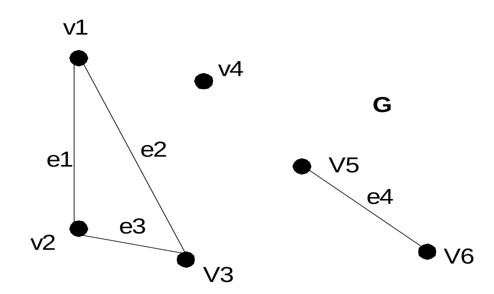
المخطط G يعتبر متصلاً اذا اعطيت اية نقاط v و w فيه فانه يوجد مسار من v الى w .

A graph G is connected if given any vertices v and w in G, there is a path from v to w

وبعبارة اخرى هو المخطط الذي فيه كل زوج من النقاط مربوط بحافة edge

A graph is connected if there is a path connecting every pair of vertices

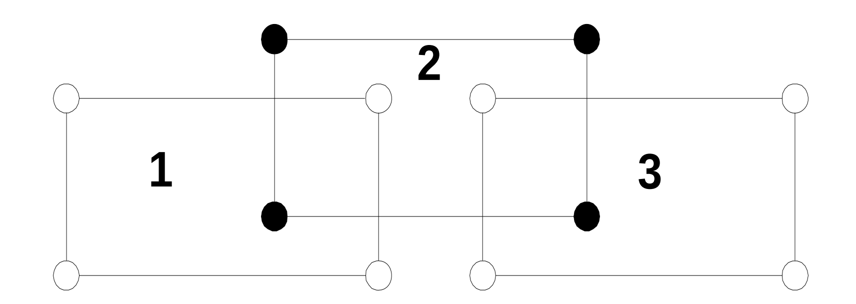
ان المخطط الموضح في الشكل (7-5) يعتبر مخطط متصل بسبب اذا اعطيت اية نقاط v و w فيه فانه يوجد مسار من v الى w . كما ان المخطط يشكل قطعة واحدة . بينما المخطط في الشكل (7-6) يكون غير متصل (*not connected*) لانه مثلاً لايوجد مسار بين النقطة v_2 و v_5 . كما انه مكون من عدة اجزاء (**pieces**) ان هذه القطع او الاجزاء هي مخططات جزئية (***sub graphs***) من المخطط الاصلي وتدعى بمكونات او عناصر المخطط (**Components**).



شكل (7-6) يبين مخطط غير متصل (***not connected***)

ومن جانب اخر ان المخطط غير المتصل (*not connected graph*) بالامكان تقسيمه الى اجزاء متصلة *connected* (*components*) او ماتسمى مخططات جزئية متصلة منفصلة (*disjoint connected subgraphs*) , على سبيل المثال , المخطط التالي مكون من ثلاثة اجزاء متصلة:

:Graph is made of three connected components .



تعريف 7-4 :

افرض ان $G=(V,E)$ تمثل مخططاً (Graph) , فاننا نسمي (V',E') مخطط جزئي (subgraphs) من G اذا:

1- $(V'\subseteq V)$ و $(E'\subseteq E)$.

2- لاي حد (edge) $(e'\in E')$, اذا e' تقع على v' و w' , فان v' و w' تنتمي الى V'

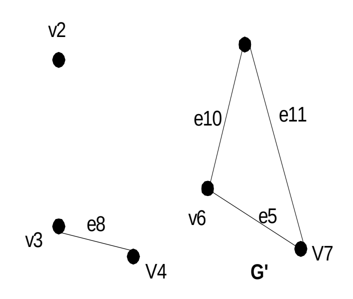
Definition : let $G=(V,E)$ be a graph , we call (V',E') a subgraphs of G if:

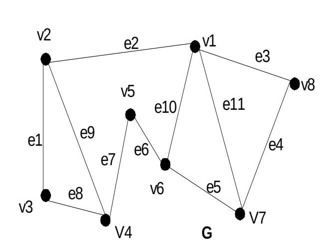
1- $(V'\subseteq V)$ and $(E'\subseteq E)$

2- For every edge $(e'\in E')$, if e' incident on v' and w' , then v' and $w'\in V'$

مثال 7-3:

المخطط $G'=(V',E')$ من الشكل (7-7) هو مخطط جزئي من المخطط $G=(V,E)$ الشكل (7-8) وذلك بسبب ان $(V'\subseteq V)$ و $(E'\subseteq E)$.



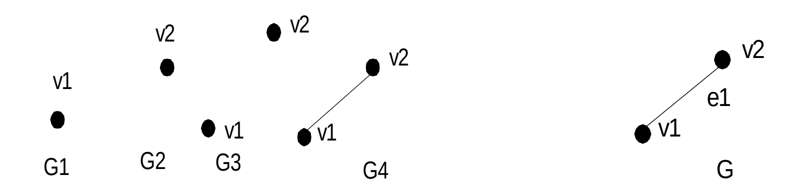


شكل (7-7) مخطط جزئي من المخطط في الشكل (7-8)

شكل (7-8) مخطط جزء منه مبين في الشكل (7-6)

مثال 7-4:

اوجد كل المخططات الجزئية (**subgraphs**) للمخطط G المبين في الشكل (7-9) لها على الاقل نقطة واحدة (***one vertex***).اذا لم نختر اي حد (edge) , فيمكننا ان نختار نقطة واحدة او كلاهما وبذلك نحصل على المخططات الجزئية G_1 , G_2 و G_3 انظر الشكل (7-10) , واذا اخترنا e_1 فانه يجب ان نختار نقطتين عليهما يقع e_1 وفي هذه الحالة نحصل على G_4. اذن G يحوي اربعة مخططات جزئية .



شكل (7-10) المخططات الجزئية للمخطط في الشكل (7-8)   شكل (7-9) مخطط المثال (7-4)

تعريف 7-5:

افرض ان **G** مخطط وافرض ان v هي نقطة في **G** .ان المخطط الجزئي G' لـ **G** الذي يحوي كل الحدود (***edges***) والنقاط (***vertices***) في **G** والتي تكون متضمنة في بعض المسار مبتدأة في النقطة v يدعى عنصراً لـ **G** يحوي v .

مثال 7-5:

ان المخطط **G** في الشكل (7-5) مكون من جزء واحد , بالتسمية هو نفسه, حيث في الحقيقة ان المخطط متصل اذا كان يشمل على جزء واحد (***one componen*t**).

مثال 7-6:

افرض ان لدينا المخطط في الشكل (7-6) . فان جزء المخطط **G** الذي يحتوي على v_3 هو المخطط الجزئي (***subgraph***) التالي:

$$G_1 = (V_1, E_1), \qquad V_1 = \{v_1, v_2, v_3\}, \qquad E_1 = \{e_1, e_2, e_3\}$$

وان جزء المخطط **G** الذي يحتوي على v_4 هو المخطط الجزئي:

$$G_2 = (V_2, E_2), \qquad V_2 = \{v_4\}, \qquad E_2 = \phi$$

وان جزء المخطط **G** الذي يحتوي على v_5 هو المخطط الجزئي:

$$G_3 = (V_3, E_3), \qquad V_3 = \{v_5, v_6\}, \qquad E_3 = e_4$$

من جانب اخر ان هناك ميزة اخرى (*another characterization*) لاجزاء او مكونـات المخطـط $G = (V, E)$ تحصـل بواسطة تعريف علاقة (*relation*) , **R** , على مجموعة من النقاط **V** بواسطة القانون (***Rule***) التالي:

$$v_1 R v_2 \qquad \text{If there is a path from } v_1 \text{ to } v_2$$

مثال 7-7 :

افرض ان **G** مخطط . عرِّف علاقة **R** على مجموعة النقاط **V** للمخطط **G** , كالعلاقة vRw , اذا كان هنـاك مسـار مـن v الى w . برهن ان **R** هي علاقة متكافئة (*an equivalence relation*) على **V**. (**راجـع العلاقـات (Relations) في الفصل الثالث).**

الحل:

اذا كانت ***v*** هي نقطة في **V** فان المسار المتضمن لـ *v* وبدون حدود (**without edges**) هو المسـار مـن ***v*** الى ***v*** , وبالتـالي فان vRv لكل نقطة ***v*** هي في **V** . وعلى اساس ذلك فان **R** علاقة مرتدة على نفسها تدعى (**Reflexive**).

افرض العلاقـة vRw . فانـه يوجـد مسـار مـن v_0 الى v_n $(v_0, \ldots\ldots v_n)$, حيـث $(v_0 = v)$ و $(v_n = w)$. والان $(v_n, \ldots\ldots v_0)$ هو المسار من w الى v , وبالتالي نحصل على العلاقة wRv , وعـلى اسـاس ذلـك فـان R هـي علاقة متماثلة (***Symmetric relation***).

من ناحية اخرى افرض العلاقتين vRw و wRx . اذن هناك مسار P_1 من *v* الى *w* والمسار P_2 من *w* الى *x* . والان P_2 متبوعاً بـ P_2 هو المسار من *v* الى *x* , وبالتالي نحصل على العلاقة vRx . وعـلى اسـاس ذلـك فـان **R** هـي علاقـة انتقالية (**Transitive**) . وبما ان **R** هـي مرتـدة عـلى نفسـها (**reflexive**), متماثلـة (**symmetric**) , و علاقـة انتقاليـة (**transitive**) . اذن هي علاقة متكافئة (**equivalence relation**) على المجموعة V .

وبعد اثبات ان العلاقة R هي علاقة متكافئة في المثال اعلاه, فانه اذا كانت $v \in V$, فان مجموعة النقاط في الجـزء الـذي يحوي *v* هو الفئة المكافئة (**Equivalence class**) التالية:

$$[v]=\{w\in V \mid wRv\}$$

لاحظ ان تعريف المسار يسمح بتكرار النقاط او الحدود , على سبيل المثال: المسار(1.2.6) في الشكل (7-5) ربما يكتب $(1,2,3,4,2)$ كما ذكرنا سابقاً. اي النقطة 2 تكررت مرتين . من جانب اخر فان الفئات الجزئية (*subclasses*) تحصل بواسطة منع (*prohibiting*) حصول مضاعفة (*duplicate*) النقاط او الحدود او بجعل النقاط v_0 و v_n في التعريف (7-2) متماثلتين (*identical*).

تعريف 7-6

افرض ان v و w هما نقاط في المخطط G . فالمسار البسيط (*Simple Path*) من v الى w هو المسار من v الى w بدون تكرار النقاط (*with no repeated vertices*).

والحلقة اوالدورة الكاملة (*cycle*) او (*circuit*) هي المسار لطول يساوي صفراً (*nonzero length*) من v الى w وبدون تكرار الحدود (*with no repeating edges*).

A circuit is a path which ends at the vertex it begins (so a loop is an circuit of length one).

اما الحلقة او الدورة البسيطة (*simple cycle*) فهي الحلقة من v الى v التي فيها (ماعدا نقاط البداية والنهاية اللتان مساويتان لـ v) , انه لاتوجد نقاط متكررة (*no repeated vertices*).

مثال 7-8:

لو اخذنا الشكل (7-5) [1], فاننا نحصل على المعلومات التالية , جدول (7-2):

path	*Simple path ?*	*cycle*	*Simple cycle?*
(6,5,2,4,3,2,1)	No	No	No
(6,5,2,4)	Yes	No	No
(2,6,5,2,4,3,2)	No	Yes	No
(5,6,2,5)	No	Yes	Yes
(7)	Yes	No	No

جدول (7-2) يبيّن المسار البسيط والدورة (**cycle**) والدورة البسيطة للمخطط الموضح بالشكل (7-5)

يمكننا ان نقول ان مسلكاً مغلقاً (closed walk) في مخطط (graph), هو

المسلك $v_0 e_1 v_1 e_2 \ldots\ldots\ldots v_{n-1} e_n v_0$ الذي يبدأ وينتهي عند نفس النقطة (vertex). ان المسلك المغلق يدعى دورة (cycle) اذا كانت $v_i \neq v_j$ و $e_i \neq e_j$ لكل **i** و **j** منفصلتين (*for each distinct i and j*).

مثال 7-9:

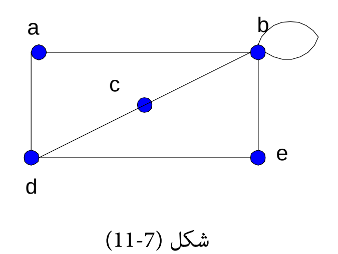
بيّن فيما اذا كان المسار المُعطى في المخطط التالي المُبين في الشكل (7-11) هو :

أ- مسار بسيط (*simple path*).

*a **simple graph** is a (usually finite) set of vertices V and set of unordered pairs of distinct elements of V called edges.*

ب- حلقة او دورة (*cycle*)

ت- حلقة بسيطة (*a simple cycle*)



شكل (7-11)

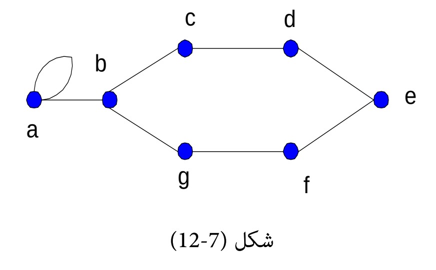
1- (b,b) وهنا يمثل حلقة (*cycle*), وحلقة بسيطة (*simple cycle*)

2- (d,c,b,e,d) وهنا يمثل ايضاً حلقة (*cycle*), وحلقة بسيطة (*simple cycle*)

3-(a,d,c,b,e) وهنا يمثل مسار بسيط (*simple path*).

مثال 7-10:

بيّن كل الحلقات البسيطة (*all simple cycles*) في المخطط التالي, شكل (7-12) :



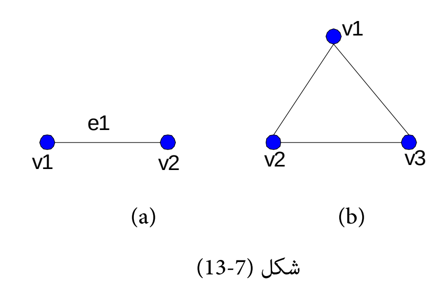
شكل (7-12)

ان كل الحلقات البسيطة (*all simple cycles*) في المخطط هي:

$$(a,a),(b,c,g,b),(b,c,d,f,g,b),(b,c,d,e,f,g,b),(c,g,f,d,c),(c,g,f,e,d,c),(d,f,e,d)$$

مثال 7-11:

اوجد كل المخططات الجزئية التي لها على الاقل نقطة واحدة للمخططات التالية, شكل(7-13) :



شكل (7-13)

المخططات الجزئية في (a) هي :

$$G_1 = (\{v_1\}, \phi)$$
$$G_2 = (\{v_2\}, \phi)$$
$$G_3 = (\{v_1, v_2\}, \phi)$$
$$G_4 = (\{v_1, v_2\}, \{e_1\})$$

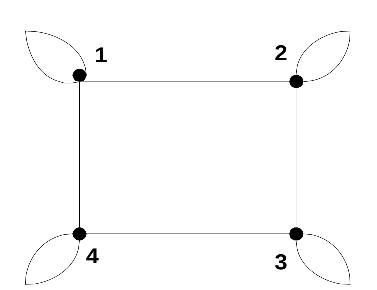
المخططات الجزئية في (b) هي: وهي (17) مخططاً جزئياً , يمكننا استنتاجها بنفس الاسلوب في **(a)** .

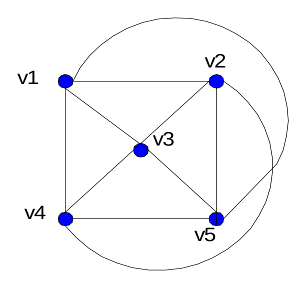
مثال 7-12:

ان درجة او مرتبة النقطة v (*degree of vertex v*) هي عدد الحدود الواقعة على *v* (*incident on v*) .

The degree (or valence) of a vertex is the number of edge *ends* at that vertex

اذن المرتبة او الدرجة لكل **نقطة** في المخطط ادناه, شكل (7-14) , هي **اربعة** بسبب ان على كل نقطة توجد **اربعة حدود**.
"*in this graph all of the vertices have degree* four".





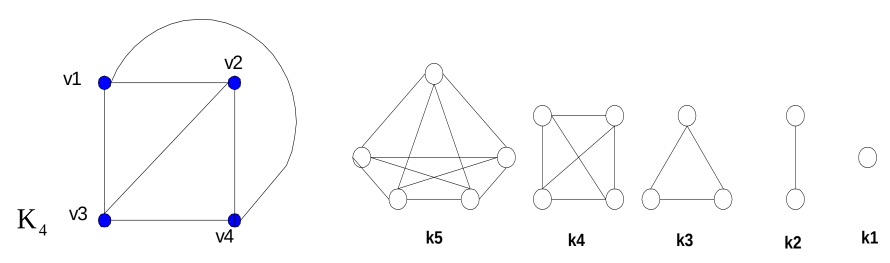
شكل (7-14) يبين المرتبة او الدرجة لكل نقطة

ان النقطة ذات الدرجة 0 تعني انها غير مربوطة بحد edge مع غيرها , اي انها معزولة isolated

A vertex of degree zero (with no edges connected) is isolated.

من جانب اخر ان المخطط الكامل (*complete graph*) على ***n*** من النقاط يرمز له بـ K_n , هو المسار البسيط مع ***n*** مـن النقاط الذي فيه يوجد حد (*edge*) بين كل زوج من النقاط المحددة (*distinct vertices*).

لذا فان المخطط الكامل على النقاط الاربعة (المخطط اليسار) يرمز لـه بـ K_4 , وكذلك المخططات الخمسة الاولى الكاملة (k1 الى k5) , انظر الشكل(7-15) ادناه :



here the first five complete graphs

شكل (7-15) يبين المخططات الكاملة(*complete graphs*)

تعريف7-6 :

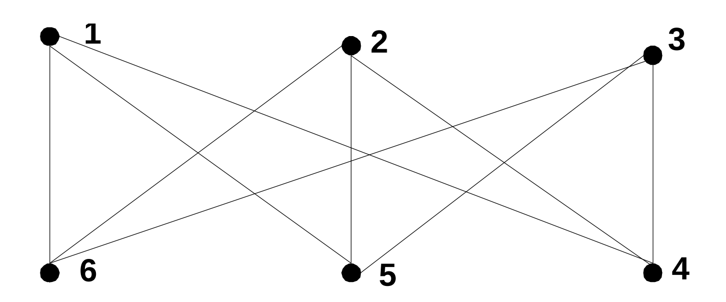
ان المخطط $G=(V,E)$ هو مقسّم ثنائي (*bipartite*) اذا وجدت مجموعتان جزئيتان (*subsets*) , V_1 و V_2 (اما من المتحمل ان يكونا خاليتين) لـ V على ان :

$$V_1 \bigcup V_2 = V \text{ و } V_1 \bigcap V_2 = \phi$$

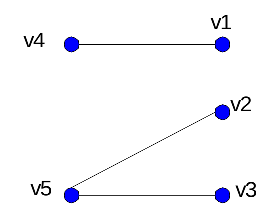
وكل حد في E يسقط على نقطة واحدة في V_1 وواحدة في V_2.

مثال 7-13:

*A graph is **bipartite** if its vertices can be partitioned into two disjoint subsets U and V such that each edge connects a vertex from U to one from V. A bipartite graph is a **complete bipartite** graph if every vertex in U is connected to every vertex in V. If U has n elements and V has m, then we denote the resulting complete bipartite graph by **Kn,m**. The following illustration shows **K3,3.***



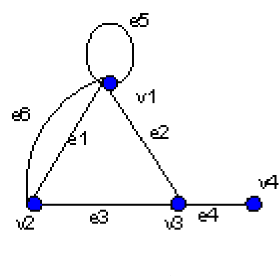
ومثال ثاني : ان المخطط في الشكل (7-16) هو قسمين **(Bipartite)** بسبب اذا فرضنا ان: $V_1 = \{v_1, v_2, v_3\}$ وان $V_2 = \{v_4, v_5\}$. نلاحظ ان حداً واحداً على كل نقطة على V_1 ومثلها على V_2.



شكل(7-16) مخطط ثنائي (*bipartite graph*)

مثال 7-14:

للمخطط $G=(V,E)$ المبين بالشكل (7-17) , اوجد E ,V , كل الحدود المتوازية (*all parallel edges*) , كل الحلقات المغلقة (*all loops*), كل النقاط المعزولة (*all isolated vertices*) , وكذلك بيِّن فيما اذا كان المخطط G هو مخطط بسيط ثم بيِّن على اية نقطة يقع الحد e_1.



الشكل (7-17)

الحل:

$V_1=\{v_1,v_2,v_3,v_4\}$, $E_1 = \{v_1, v_2, v_3, v_3\}$, كما ان e_1 و e_6 هما حدود متوازية , و e_5 هيو حلقة مغلقة (***loop***), كذلك لاتوجد نقاط معزولة , واخيراً e_1 يسقط على v_1 و v_2.

7-4 خوارزمية ايجاد المسار الاقصر بين نقطتين s و v : Shortest Path Algorithm

هذه الخوارزمية المشار اليها في الجزء (7-4-2) ادناه , هي لحل مسألة المسار الاقصر لمخطط متجه (*directed graph*) , **G** , له حدود (*edges*) بقيم موجبة (*nonnegative edge weights*). ان مدخلات الخوارزمية تشتمل على مخطط متجه ونقطة كمصدر (نقطة البداية) مثل a او s في ذلك المخطط.حيث ان **V** ترمز الى كل النقاط (*Vertices*) في المخطط **G** وان كل حد (*edge*) للمخطط ممثَّل في زوج مرتب من النقاط (u,v) يمثل اتصال من النقطة **u** الى النقطة **v** .

من ناحية اخرى , ان مجموعة كل الحدود يرمز لها بـ **E** , و قيم تلك الحدود تعطى بدالة القيم (*weight function*), w, $w : E \rightarrow [0, \infty)$, وعلى اساس ذلك تكون $w(u,v)$ هي تمثل الحركة من النقطة **u** الى المقطة **v** . ان قيمة الحد يمكن ان نفكر بها على انها كالمسافة بين النقطتين . اما قيمة المسار فانها تمثل مجموع قيم الحدود في ذلك المسار , وان اقل قيمة للمسار تمثل المسار الاقصر , على سبيل المثال : اذا اعطينا نقطتين هما **s** و **t** في **V** , فان الخوارزمية تجد المسار من **s** الى **t** باقل قيمة (*lowest cost*) التي هي المسار الاقصر (*the shortest path*). وبالطبع يمكن ان نستخدم الخوارزمية لايجاد المسار من اية نقطة كمصدر الى اية نقطة في المخطط **G** .

7-4-1 وصف الخوارزمية: Algorithm Description

ان الخوارزمية المذكورة تعمل بابقاء قيمة $d[v]$ تمثل المسار الاقصر بين **s** و **v** , حيث ان القيمة الابتدائية لها صفراً $d[s]=0$ للنقطة المصدر (نقطة البداية) وهي تمثل هنا النقطة s , وتكون مالانهاية ∞ (*infinity*) لبقية النقاط ماعدا **s** , اي اننا لانعلم اي مسار يقودنا الى تلك النقاط : ($d[v]= \infty$ **for every** v **in** V**, except** s) . وعندما تنتهي الخوارزمية , فان $d[v]$ هي قيمة المسار الاقصر من **s** الى **v** . او تكون تساوي مالانهاية اذا كان المسار غير موجود. ان الخوارزمية تحافظ على مجموعتين (*two sets*) من النقاط **S** و **Q** . ان المجموعة **S** تمثل كل النقاط التي لها نعرف ان القيمة $d[v]$ تمثل المسار الاقصر , بينما **Q** تمثل بقية النقاط. ان المجموعة S تبدأ مجموعة فارغة وبعد كل خطوة تنفيذ للخوارزمية ترحل نقطة واحدة من **Q** الى **S** . ان تلك النقطة تختار كنقطة باقل قيمة لـ $d[v]$, وعندما النقطة **u** تنتقل الى **S** , فان الخوارزمية تحذف كل حد ملغي (*outgoing edge* (u,v)) من المخطط .

7-4-2 شفرات الخوارزمية الشبيهة بشفرات برنامج الحاسوب -Pseudocodes

كما ذكرنا اعلاه ,فان الخوارزمية التالية هي خوارزمية ايجاد المسار الاقصر بين النقطتين **s** و **v** في المخطط المتجه **G** .

```
1   function Dijkstra(G, w, s)
2   for each vertex v in V[G]          // Initializations
3   d[v] := infinity                   // Unknown distance function from s to v
4   previous[v] := undefined
5   d[s] := 0                          // Distance from s to s
6   S := empty set                     // Set of all visited vertices
```

```
7   Q := V[G]                        // Set of all unvisited vertices
8   while Q is not an empty set      // The algorithm itself
9   u := Extract_Min(Q)              // Remove best vertex from priority queue
10  S := S union {u}                 // Mark it 'visited'
11  for each edge (u,v) outgoing from u
12  if d[u] + w(u,v) < d[v]    // Relax (u,v)
13  d[v] := d[u] + w(u,v)
14  previous[v] := u
```

في الخوارزمية اعلاه وبالتحديد في السطر 9: ان u := Extract_Min(Q) تبحث عن النقطة u في مجموعة النقاط Q التي لها اقل قيمة d[u]. ان تلك النقطة يتم ازالتها من المجموعة Q واعادتها الى مستخدم الخوارزمية. واذا كنا نريد ايجاد المسار الاقصر بين s و t . فانه بالامكان ايقاف البحث عند السطر 9 اذا كانت $(u = t)$. والان يمكننا قراءة المسار الاقصر من s الى t بجمل التكرار التالية :

```
1 S := empty sequence
2 u := t
3 while defined previous[u]
4       insert u to the beginning of S
5       u := previous[u]
```

ان المجموعة **S** تمثل كل النقاط التي تشتمل على واحد من المسارات القصيرة من **s الى t** او تمثل مجموعة فارغة اذا لم يوجد مسار.

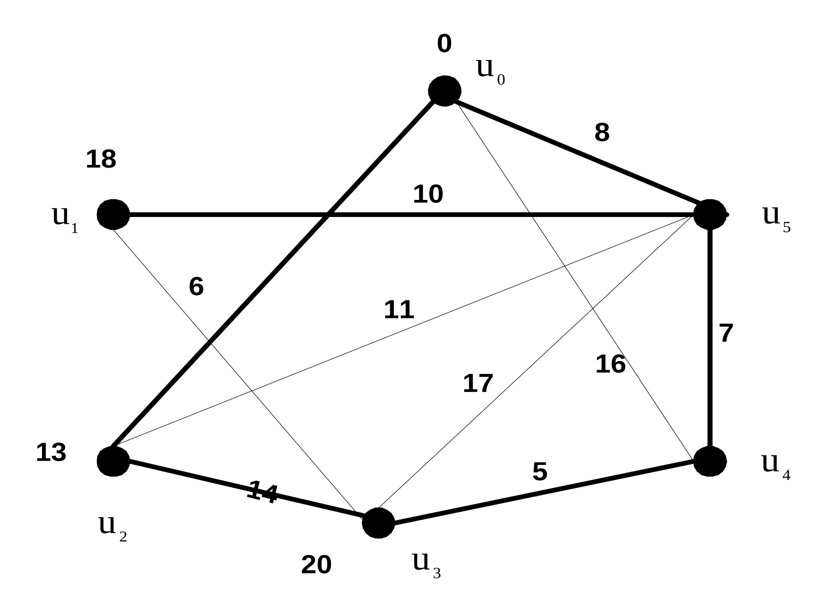
ان المسألة العامة والشاملة هي لايجاد كل المسارات الاقصر بين s و t (حيث من الممكن ان يوجد عدة مسارات لها نفس الطول) .لذا بدلاً من من تخزين نقطة واحدة في كل ادخال للمصفوفة **previous**[] اعلاه , فاننا يجب ان نخزن كل النقاط التي تحقق الشرط في السطر **12** في الخوارزمية (*relaxation condition*).

على سبيل المثال: اذا كان كل من **r و s** متصلين الى t وكل منهما يقع على مسارات اقصر مختلفة خلال **t** (اي قيمة الحد (*edge*) هو نفسه في كلتا الحالتين) . اذن يجب ان نضيف كل من r و s الى المصفوفة **previous[t]** . وعندما تنتهي الخوارزمية , فان تركيبة بيانات

المصفوفة (previous[] data structure), تصف بشكل حقيقي مخطط هو مخطط جزئي (*subgraph*) للمخطط الاصلي بحذف بعض الحدود.

مثال 7-15:

في المخطط ادناه , جد المسار الاقصر من u_0 ؟



باستخدام خوارزمية المسار الاقصر المذكورة انفاً يمكننا ايجاد المسار الاقصر من u_0 كمايلي:

1- نبدأ من النقطة u_0 ويكون طول المسار صفراً, ولبقية النقاط تكون المسافة غير معروفة ∞ , ومن هذه النقطة سيتم اختيار النقاط المجاورة لها والمتصلة معها بحد (edge) كما هو مبين في المخطط. وهذه النقاط هي u_5 وتجد الخوارزمية بُعدها او مسافتها عن u_0 وستكون على بُعد 8 , u_4 على بُعد 16, و u_2 على بعد 13 , ونتيجة لذلك سيتم اختيار اقرب نقطة وهي u_5 مع الاحتفاظ بالنقاط السابقة وابعادها.

2- في الخطوة التالية نبدأ من u_5 لانها الاقرب الى u_0 وناخذ النقاط المجاورة لها وهي u_1 وتحسب مسافتها وستكون 18(8+10) انظر المخطط . والنقطة u_4 على مسافة

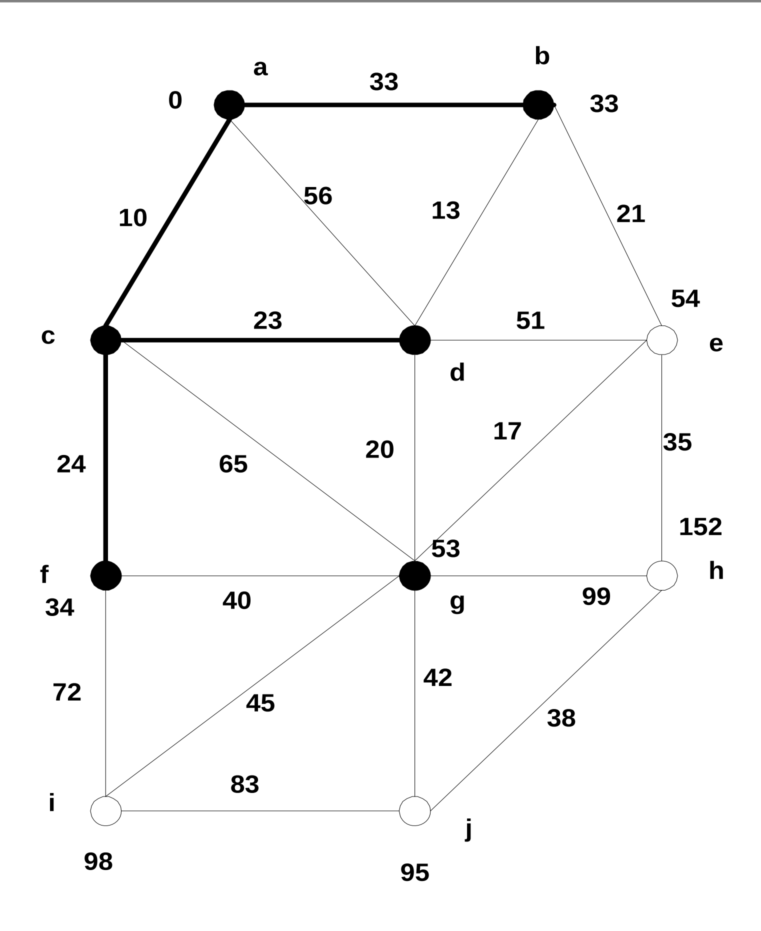
15(8+7) , و u_3 ستكون على مسافة 25(17+8). اما النقطة u_2 فتبقى u_2 على مسافة 13 , حيث المسافات المحسوبة جميعها من u_0 الى النقاط المذكورة اعلاه.

3- في هذه الخطوة يتم اختيار u_2 على مسافة 13مع الابقاء على نقاط تجاور u_5.

4- في الخطوة التنالية يتم اختيار u_4 على مسافة 15(8+7) مع الغاء الحد بين u_5 و u_3 ورسم الحد بين u_5 و u_4 على مسافة 15 من u_0, وبين u_4 و u_3 حيث ستكون u_3 على مسافة 20 من u_0 (8+7+5) هذه المرة.

5- وهنا يتم اختيار u_1 على مسافة 18 من u_0 (8+10). واخيرا يتم اختيار u_3 بوضع علامة عليها وتكون على بعد 20 , ونحصل على المسارات القصيرة وهـي مـن u_0 الى u_5 الى u_4 الى u_3 (8+7+5). ومـن u_0 الى u_5 الى u_1 وهي (8+10) , ومن u_0 الى u_2 وهي 13 .

والاسلوب اعلاه يمكن تطبيقه بسهولة على المخطط التالي ونترك ذلك للقارئ.



ويمكننا التعبير عن الخوارزمية السابقة بالخوارزمية التالية بشكل اوضح مع امثلة تطبيقية عليها , انظر الجزء (7-5) التالي:

5-7 تطبيق خوارزمية ايجاد المسار الاقصر A Shortest-Path Algorithm Application

لقد تطرقنا فيما سبق الى المخططات التي فيها تعًين قيم موجبة للحدود (nonegative edge weights) والتي تسمى (*weighted graphs*), فان طول المسار فيها هو مجموع قيم الحدود في المسار. نحن افترضنا ان $w(i, j)$ ترمز الى قيمة الحد (i, j) (*a weight of the edge* (*i*, *j*)). ان الهدف في تلك المخططات (*weighted graphs*) هو ايجاد المسار الاقصر (*shortest path*) (يعني المسار الذي له اقل طول *minimum length*) بين نقطتين معطاة في المخطط.

ان الخوارزمية المبينة ادناه نسبة الى عالم الحاسوب الالماني (E.W.Dijkstra) حلّت تلك المسالة بكفاءة. وفي هذاالجزء نرمز الى المخطط **G** بانه مخطط متصل وحدوده معطاة لها قيماً (*Connected and weighted graph*) , ونفترض ان قيم الحدود هي اعداد موجبة ونريد ان نجد المسار الاقصر من النقطة ***a*** الى النقطة ***z*** .

تتضمن خوارزمية E.W.Dijkstra اعطاء علامات(*labels*) الى النقاط (*vertices*) , ونفرض ان **L(v)** يرمز الى علامة النقطة ***v*** . وفي اية نقطة (مرحلة) في تنفيذ الخوارزمية , فان بعض النقاط (*vertices*) يكون لها علامات مؤقتة *temporary* (*labels*) والباقي يكون له علامات دائمة (*permanent labels*).

نفترض ان **T** تمثل مجموعة النقاط التى لها علامات مؤقتة , وفي توضيح الخوارزمية فاننا نحيط النقاط (*Vertices*) التي لها علامات دائمة بدائرة (*circle*) . وسنبين لاحقاً انه اذا **L(v)** هي علامة دائمة للنقطة ***v*** , فان **L(v)** هي المسار الاقصر من ***a*** الى ***v*** . من جانب اخر , فانه في البداية ان كل النقاط لها علامات مؤقتة . ثم ان في كل اعادة (*iteration*) للخوارزمية تتغير حالة علامة واحدة من المؤقتة الى الدائمة , وبالتالي فان الخوارزمية تتوقف (*terminate*) عندما ***z*** تستلم العلامة الدائمة . وعند تلك الحالة فان **L(z)** يعطي المسار الاقصر من ***a*** الى ***z***.

7-5-1 خوارزمية Dijkstra للمسار الاقصر

Dijkstra's Shortest Path Algorithm

هذه الخوارزمية تجد طول المسار الاقصر من النقطة **a** الى النقطة **z** في مخطط متصل (***connected graph***) وحدوده لها قيماً (***weighted edges***) . ان قيمة الحد (i, j) هي $[w(i, j) > 0]$ وعلامة النقطة ***x*** هي **L(*x*)** . وعند انتهاء تنفيذ الخوارزمية فان **L(z)** يمثل المسار الاقصر من ***a*** الى ***z***. وفيمايلي الخوارزمية [1]

المدخلات ***inputs*** : مخطط متصل وحدوده لها قيماً عبارة عن اعداد موجبة , ونقاط **a** و **z**

المخرجات ***outputs*** : المسار **L(z)** , المسار الاقصر بين النقطة **a** و **z** .

وفيمايلي خطوات الخوارزمية بشكل شفرات شبيهة(Pseudocodes) بشفرات البرنامج الحقيقية:

```
1.  function dijkstra(w, a, z, l)
2.  L(a)=0                                  // distance from a to a
3.  for all vertices  x ≠ a  do             // Initializations
4.  L(x) := ∞                               // unknown distance  function from x to a
5.  T := set   of   all   vertices
6.  // T is the set of vertices whose shortest distance from a has not been found
7.  while  z ∈ T  do
8.  begin
9.  choose  v ∈ T  with minimum  L(v)
10. T := T − {v}
11. for each  x ∈ T  adjacent to v do // to choose adjacent vertices to circled vertex
12. L(x) := min{L(x), L(v) + w(v, x)}
13. end
14. end function dijkstra
```

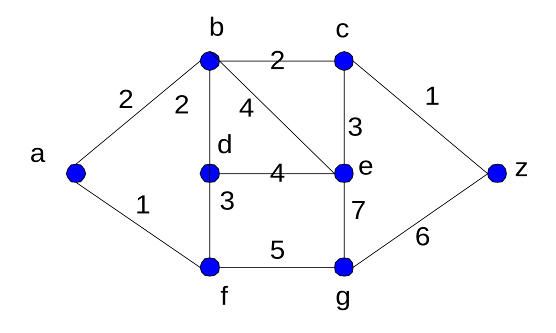
مثال 7-16:

في هذا المثال نبيِّن كيف ان خوارزمية *Dijkstra* تجد المسار الاقصر مـن **a** الى **z** في المخطـط المبـين بالشـكل (7-18)[1] . ان النقاط في **T** لم تحدد بدوائر ولها علامات مؤقتـة(*temporary labels*) .النقـاط ذات العلامـة الدائريـة لهـا علامـات دائمـة .الشكل (7-19) يبين نتيجةتنفيذ الخطوات 2-5 في الخوارزمية اعلاه.في السـطر 7 ان **z** لـيس لهـا علامـة . ويسـتمر الاجـراء لغاية السطر **9** حيث اخترنا النقطة **a** التي هي غير محاطة بدائرة وتمثل اصغر علامة , ونضع دائرة عليها انظر الشـكل(7-20) . في الاسطر **11** و **12** عملنا تحديث لكل من النقاط **b** و *f* غير المحاطة بدائرة والمجاورة لـ **a** .

$$L(b) = min\{\infty, 0 + 2\} = 2$$

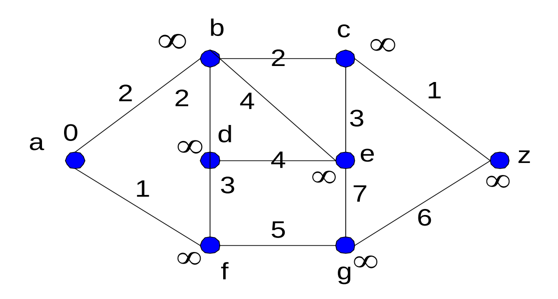
$$L(f) = min\{\infty, 0 + 1\} = 1$$

انظر الشكل(7-20) ونحن هنا في المرة الاولى (*first iteration*) لتنفيذ الخوارزمية .



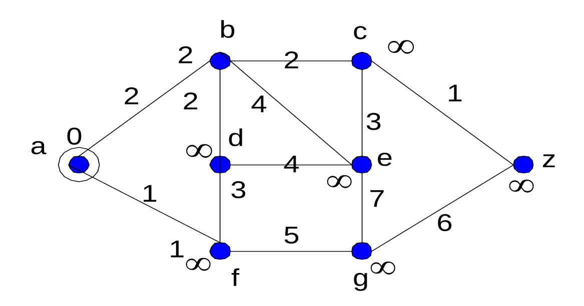
شكل(7-18) مخطط المثال (7-16)

وعند هذه النقطة نعود الى السطر 7.

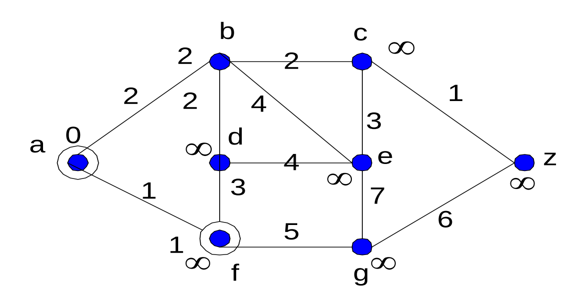


شكل(7-19)اعطاء القيم الاولية في خوارزمية المسار الاقصر لـ *Dijkstra*

بما ان النقطة **z** لم تحط بدائرة , فنستمر الى السطر **9** , حيث نختار النقطة ***f*** النقطة غير المحاطة بدائرة مع العلامة الاصغر , فنوضع عليها دائرة. انظر الشكل (7-21), في الاسطر **11 و 12** نحدث كل علامة للنقاط غير المحاطة بدائرة وهي **d** و **g** المجاورة لـ ***f***. ونحصل على العلامات (***labels***) المبينة بالشكل (7-21) ونحن هنا في التكرار الثاني (*second iteration*) لتنفيذ الخوارزمية اعلاه.

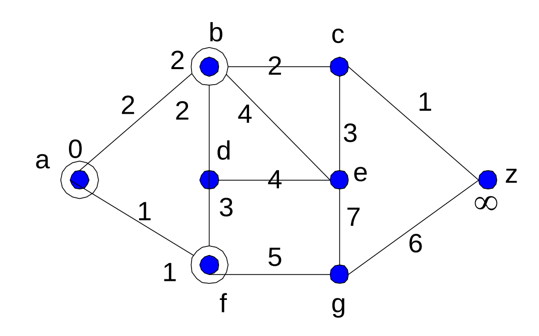


شكل(7-20) يمثل التنفيذ الاول لخوارزمية *Dijkstra*



شكل(7-21)) يمثل التنفيذ الثاني لخوارزمية *Dijkstra*

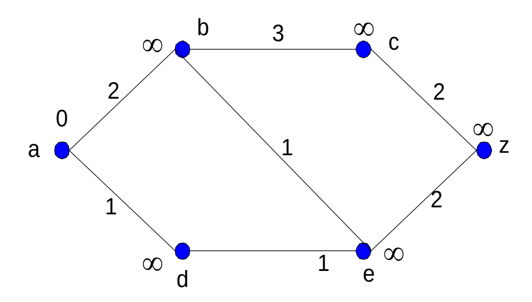
وهنا يجب ان نثبت ان التنفيذ القادم (الثالث) للخوارزمية ينتج العلامات المبينة في الشكل(7-22) وهو يمثل انتهاء تنفيذ الخوارزمية . ان **z** قد اعطيت العلامة 5 التي تبين ان طول المسار الاقصر من **a** الى **z** هو **5**. ان المسار القصر يعطى بواسطة (**z, c, b, a**).



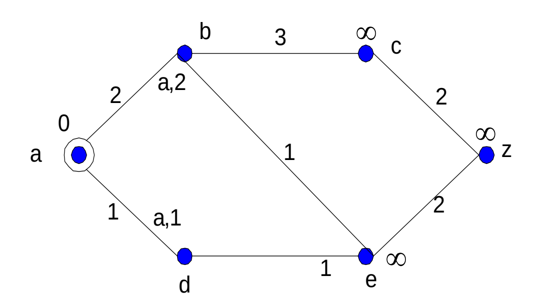
شكل(7-22)) يمثل التنفيذ الثالث والاخير لخوارزمية *Dijkstra*

مثال 7-17 :

اوجد طول المسار الاقصر من **a** الى **z** للمخطط المبين في الشكل (7-23)

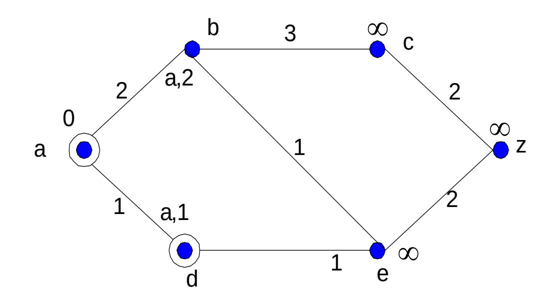


سنستخدم في ذلك خوارزمية ***Dijkstra*** التي وضحناها سابقاً[1], مع تعـديل بسـيط حيـث بالاضـافة الى وضـع دائـرة عـلى النقطة **(vertex)** , فاننا سنضع علامة عليها مع الاسم للنقطـة التـي منهـا وضـعت العلامـة . الشـكل (7-23) يبـين نتيجـة التنفيذ للاسطر من **2-4** لخوارزمية ***Dijkstra*** الانفة الـذكر . اولاً نضـع دائـرة عـلى النقطـة **a** , انظـر الشـكل (7-24) , وفي الخطوة التالية نضع علامة لكل من **b** و **d** المجاورتين لـ **a** . النقطة **b** تاخذ العلامة **(a,2)** لتبين قيمتها والحقيقة التي هي اخذها العلامة من **a** . وبشكل مشابه ان النقطة **d** تصبح **(a,1)**.



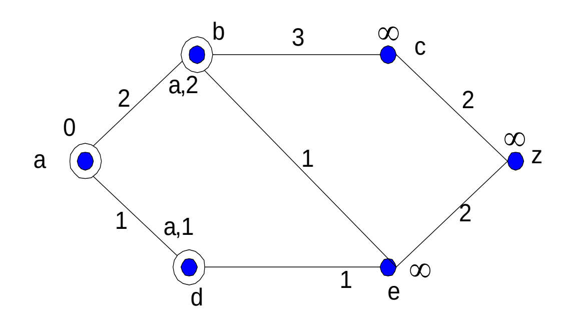
الشكل (7-24) يمثل التنفيذ الاول (*first iteration*) لخوارزمية ***Dijkstra***

ان الخطوة القادمة هي وضع دائرة على النقطة **d** وتحديث علامة النقطة **e** المجاورة الى **d** . انظر الشكل (7-25)



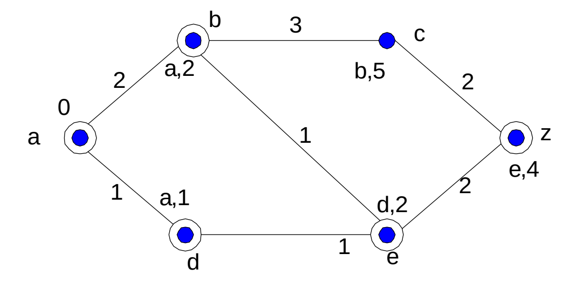
الشكل (7-25) يمثل التنفيذ الثاني (*second iteration*) لخوارزمية ***Dijkstra***

ومن ثم نضع دائرة على النقطة **b** ونحدِّث علامات (*labels*) النقاط **c** و **e** . انظر الشكل (7-26)



الشكل (7-26) يمثل التنفيذ الثالث (*third iteration*) والاخير لخوارزمية ***Dijkstra***

والخطوة القادمة نضع الدائرة على النقطة **e** ونحدِّث علامة النقطة **z** انظر شكل(7-27)



شكل(7-27) يمثل نتيجة خوارزمية المسار الاقصر لـ ***Dijkstra***

عند هذه النقطة نضع دائرة على النقطة **z** , لذا تنتهي الخوارزمية من التنفيذ. ان طول المسار الاقصر من **a** الى **z** هـو **4** .
نبدأ من **z** بامكاننا نعيد المسار على النقاط المحاطة بدوائر نجد ان المسار الاقصر هو:

(z, e ,d ,a)

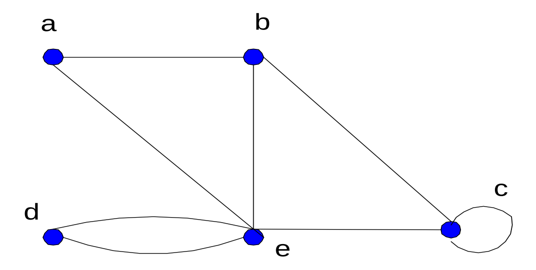
6-7 تمثيلات المخططات Representations Of Graphs

في المقاطع السابقة مثّلنا المخط بالرسم (*drawing*) . لكن بعض الاحيان , على سبيل المثال نريد استخدام الحاسوب لتحليـل المخطط , فاننا نحتاج الى صيغة اكثر ملاءمة للتمثيل . فطريقتنا الاولى لتمثيل المخططات هـي مـاتركس او مصـفوفة التجـاور (*adjacency matrix*) . ويقال ان نقطتين متجاورتين (*adjacent*) الى بعضهما اذا كان يربط بينهما حد (*edge*).

"Two vertices are adjacent if they are connected by an edge. "

مثال 7-18 :

اعتبر المخطط المبين في الشكل (7-28) .



شكل(7-28) المثال (7-18)

وللحصـول عـلى مـاتركس التجـاور لهـذا المخطـط . اولاً نختـار الترتيـب للنقـاط (*vertices*) في المخطـط , قـل مـثلاً (e,d,c,b,a). ومن ثم نضع علامات الصفوف (*rows*) والاعمدة (*columns*) للماتركس مع النقاط المرتبة .. ان قيمة عنصر الماتركس المدخلة (*entry*) في الصف **i** والعمود **j** , حيث $(i \neq j)$ هو عدد الحدود او الاضلاع (*edges*) الواقعـة على **i** و **j** . اذا كانت $(i = j)$, فان القيمة المدخلة هي ضعف عدد الحلقات (*loops*) الواقعة عـلى **i** , لاحـظ الحلقـة (***loop***) في **c** وكذلك في **e** و **d** جعلـت قـيمهم **2** في المـاتركس , واذا كتنـت **a** موصـولة بــ **b** فـان القيـمـة في المـاتركس تساوي **1** وهكذا.ان ماتركس التجاور لهذا المخطط هو:

$$
\begin{array}{c} \\ a \\ b \\ c \\ d \\ e \end{array}
\begin{array}{c} \begin{array}{ccccc} a & b & c & d & e \end{array} \\
\begin{pmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 2 & 0 & 1 \\ 0 & 0 & 0 & 0 & 2 \\ 1 & 1 & 1 & 2 & 0 \end{pmatrix} \end{array}
$$

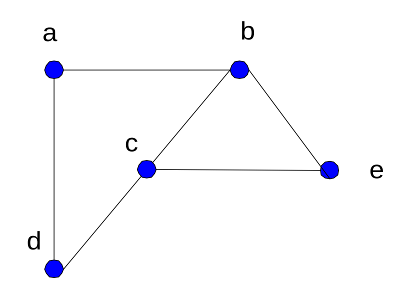
ماتركس التجاور (*adjacency matrix*)

لاحظ انه بامكاننا الحصول على درجة النقطة v (*degree of vertex v*) في المخطط **G** كما اشرنا اليها سابقاً بواسطة جمع الصف لـ **v** او العمود لـ **v** في مخطط ماتركس التجاور اعلاه, مثلاً الدرجة للنقطة **a** هي **2 (1+0+0+0+1)** ولـ **c** هي **4** ولـ**e** هي **5** ....الخ.

ان ماتركس التجاور ليس طريقة كفوءة بشكل كبير لتمثيل المخطط. وبما ان الماتركس متماثل حول القطر الاساسي *main* (*diagonal*), (يعني العناصر على الخط من الزاوية العليا اليسرى الى الزاوية اليمنى السفلى) فان المعلومات (*the information*) ماعدا الموجودة على القطر الاساسي (*main diagonal*) تبدو مضاعفة .

مثال 7-19:

ان ماتركس التجاور للمخطط البسيط المبين بالشكل (7-29) التالي:



شكل (7-29) للمثال (7-19)

هو مايلي:

$$A = \begin{matrix} & a & b & c & d & e \\ a & 0 & 1 & 0 & 1 & 0 \\ b & 1 & 0 & 1 & 0 & 1 \\ c & 0 & 1 & 0 & 1 & 1 \\ d & 1 & 0 & 1 & 0 & 0 \\ e & 0 & 1 & 1 & 0 & 0 \end{matrix}$$

ماتركس التجاور للمخطط البسيط المبين بالشكل (7-29)

سنبين انه اذا كان **A** هو ماتركس تجاور لمخطط بسيط **G** , فان الماتركسات الأُسيِّة لـ**A** (**Powers of A**) التالية :

$$A, A^2, A^3, ...$$

تحصي عدد المسارات باطوال مختلفة . وبدقة اكثر اذا كانت النقاط (*vertices*) الى المخطط **G** تم وضع علامات عليها مثل **1, 2**, ...فان القيمة المدخلة ***ij*** في الماتركس A^n تساوي عدد المسارات من ***i*** الى ***j*** بطول **n**. على سبيل المثل تصور اننا ربَّعنا الماتركس في هذا المثال, فاننا نحصل على:

$$A^2 = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{pmatrix} = \begin{matrix} & a & b & c & d & e \\ a & 2 & 0 & 2 & 0 & 1 \\ b & 0 & 3 & 1 & 2 & 1 \\ c & 2 & 1 & 3 & 0 & 1 \\ d & 0 & 2 & 0 & 2 & 1 \\ e & 1 & 1 & 1 & 1 & 2 \end{matrix}$$

اعتبر ان القيمة الداخلة في الصف **a** والعمود **c** في الماتركس A^2 حصلت من حاصل ضرب القيم الداخلة في الصف **a** في القيم الداخلة في العمود **c** في الماتركس **A** , وافرض ان:

$$\begin{matrix} & & b & & d & & c \end{matrix}$$

$$a\begin{pmatrix}0 & 1 & 0 & 1 & 0\end{pmatrix} \quad \begin{pmatrix}0\\1\\0\\1\\1\end{pmatrix}\begin{matrix}\\b\\\\d\\\\\end{matrix} = 0.0+1.1+0.0+1.1+0.1=2$$

ان القيم التي لاتساوي صفراً في المجموع اعلاه تنتج اذا كان حاصل ضرب القيم الداخلة من كل من الصف والعمود هي قيم الواحد. وهذا يحصل اذا كانت توجد نقطة (*a vertex*) (v) تكون قيمتها الداخلة في الصف **a** هي واحد ومثلها في العمود **c** . بتعبير اخر , يجب ان تكون الحدود (***edges***) بصيغة (a,v) و (v,c), مثل تلك الحدود تشكل المسار (a,v,c) بطول **2** من a الى c . وكل مسار يزيد المجموع بمقدار 1.

في هذا المثال, ان المجموع هو 2 بسبب انه يوجد مسارين : (a,b,c) و (a,d,c) بطول 2 للوصول من **a** الى **c** . انظر المخطط (7-29) في المثال (7-19), وبشكل عام ان القيمة الداخلة في الصف x والعمود y من الماتركس A^2 هي عدد المسارات بطول **2** من النقطة x الى النقطة y . ان المدخلات على القطر الرئيسة للماتركس A^2 تعطينا درجات النقاط (*the degrees of the vertices*) (عندما يكون المخطط مخططاً بسيطاً) . اعتبر على سبيل المثال , النقطة **c** ان الدرجة لـ **c** هي **3** بسبب **c** تسقط على ثلاثة حدود (*edges*) , اي ان c ترتبط بثلاثة حدود , انظر المخطط (7-29) ايضاً, التي هي: (c,b) , (c,d) , و (c,e) . لكن كل واحد من تلك الحدود يمكن تحويله الى مسار بطول **2** من **c** الى **c** :

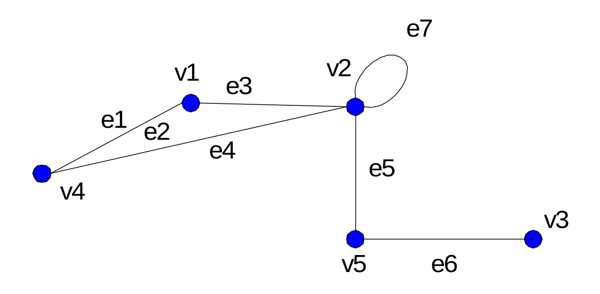
$$(c,b,c)\ ,\ (c,d,c)\ ,\ \text{و}\ (c,e,c)$$

وبشكل مشابه , ان مساراً بطول 2 من c الى c يعرِّف الحد الساقط على c. وعلى اساس ذلك فان عدد المسارات بطول 2 من c الى c هو 3 , تمثل درجة الـ c .

مثال 7-20 :

ان الطريقة الثانية لتمثيل المخططات هي ماتركس وقوع او اسقاط الحدود على النقاط (*incidence matrix*) .

من اجل الحصول على ماتركس وقوع او اسقاط الحدود على النقاط للشكل (7-30) ادناه :



شكل(7-30) يمثل مخطط المثال 7-20)

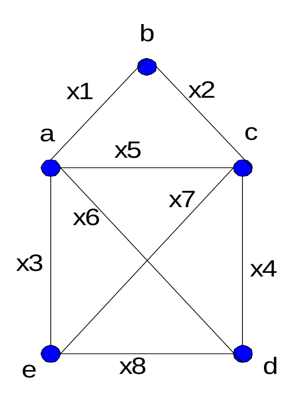
فاننا نضع علامة النقاط على الصفوف (*rows*) وعلامة الحدود (*edges*) على الاعمدة بترتيب عشوائي , وتكون مدخلات الصف ***v*** والعمود ***e*** هي **1** اذا كان ***e*** يسقط على ***v*** (*incident on v*) , وغير ذلك صفراً.وعلى اساس هذا فان ماتركس وقوع او اسقاط الحدود على النقاط هو:

$$\begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{array}{c} \begin{array}{ccccccc} e_1 & e_2 & .. & .. & .. & .. & e_7 \end{array} \\ \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{pmatrix} \end{array}$$

ونلاحظ من الماتركس ان العمود e_7 على سبيل المثال يمثل وجود حلقة (*loop*). كذلك ان العمود الذي يحوي واحدين (two 1's) لايحوي حلقة (*loop*), واخيراً فان مجموع كل صف يمثل درجة النقطة(عدد الحدود الواقعة عليها) , لاحظ مثلاً v_2 يقع عليها 5 حدود لاي ان مجموع صفّها يساوي (0+1+1+1+1+0+1 =5) وهكذا للنقاط الاخرى.

مثال 7-21 :

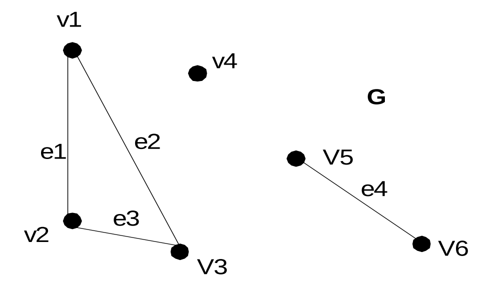
اوجد ماتركس التجاور **(Adjacency Graph)** للمخططات التالي:



ان ماتركس التجاور للمخطط اعلاه هو:

$$
\begin{array}{c} \\ a \\ b \\ c \\ d \\ e \end{array}
\begin{array}{c} \begin{array}{ccccc} a & b & c & d & e \end{array} \\
\begin{pmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{pmatrix} \end{array}
$$

وماتركس التجاور للمخطط التالي:

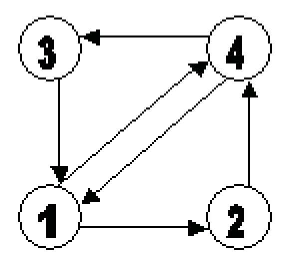


هوكالاتي:

$$\begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{array} \begin{array}{c} \begin{array}{cccccc} v_1 & v_2 & v_3 & . & . & v_6 \end{array} \\ \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \end{array}$$

تمارين الفصل السابع

1-Find a walk with five edges from the vertex 4 to vertex 2 in the following figure .is this walk a path?



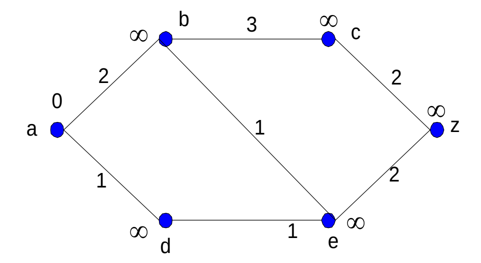
2-Draw the graphs on the set {a, b, c, d, e} with the following edge sets:

{{a}, {a, b},{b, c},{b},{c},{d},{d, e},{e}}

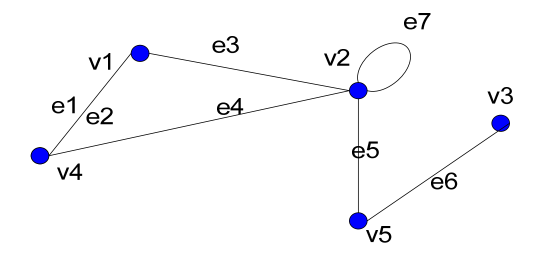
3-Find the graph of the following matrix and find the length of the paths from a to c.

$$A = \begin{array}{c} \\ a \\ b \\ c \\ d \\ e \end{array} \begin{array}{c} \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{pmatrix} \end{array}$$

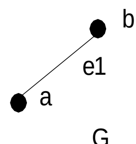
4-Find the shortest path from a to z and its length for the following graph:



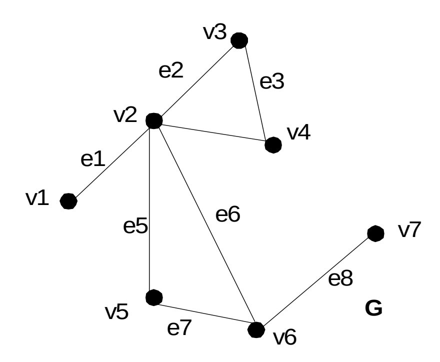
5- Write down the Incidence matrix of the following graph



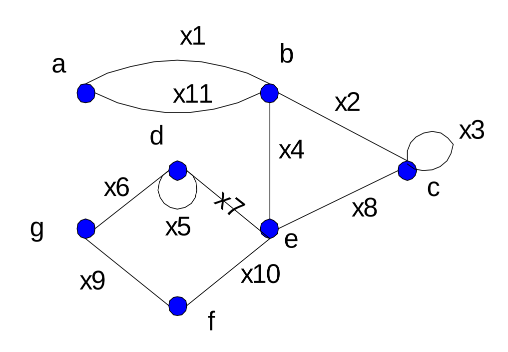
6- Find all the sub graphs of the following graph G:

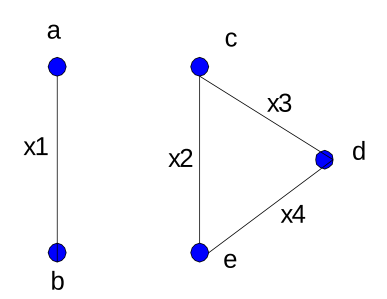


7- Find the simple path, cycle and the simple cycle of the following graph:



8- اختبر نفسك بايجاد ماتركس التجاور للمخططات التالية





الفصل الثامن

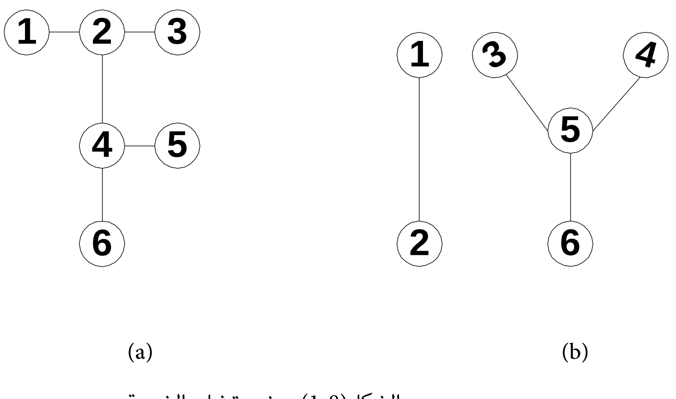
الشجرة

Tree

1-8 المقدمة

ان اول ظهور للاشجار (*trees*) كان في دراسة خواص الجزئيات في الكيمياء, وفي دراسة الفولتيات في الدوائر الالكترونية , واليوم تستخدم في طرق مختلفة في علم الحاسوب , ففكرة شجرة التفرُّع الثنائي (*binary tree*) مفيدة في دراسة القوانين الرياضية (*formulas*) , وفي دراسة القوائم المرتبة ابجدياً , وفي دراسة مايسمى بالعمليات الفرعية (*branching processes*). اما فكرة الشجرة الفرعية (*spanning tree*) فهي مفيدة جداً في سياق الاتصالات الفعالة (*context of efficient communication*) , وفي تحديد المسافات بين النقاط في انظمة النقل (*transportation systems*) وكذلك في الدراسة النظرية للمخططات (*abstract study of graphs*).

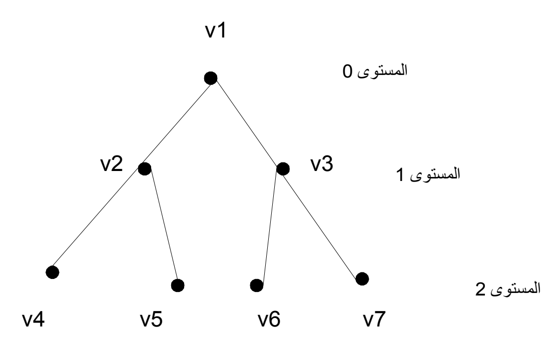
وفكرة الشجرة انها تمثل مخطط متصل (*connected graph*) لـه خاصية انه لكل حد(*edge*{x,y}), ان حذف {x,y} من مجموعة الحد *edge set* فقط دون حذف شيء من مجموعة النقاط (*vertex set*) طبعاً , يعطي مخططاً (*graph*) فيه x و y غير متصلة . وحسب التعريف فان المخطط (a) في الشكل (1-8) يمثل شجرة لان فصل اي حد (*edge*) ,على سبيل المثال الحد $\{6,4\}$, فانه يجعل المخطط غير متصل النهايات , والمخطط (b) لايمثلها لانه اصلاً مخطط غيرمتصلاً (*not connected graph*) .



الشكل(1-8) يوضح تمثيل الشجرة

وتعرَّف الشجرة **T** ايضاً بانها مخطط بسيط يحقق مايلي: اذا كانت ***v*** و ***w*** هما نقاط في الشجرة , فانه يوجد هناك مسار وحيد بين ***v*** و ***w*** . واذا شكَّلت بعض النقاط في الشجرة تصميم الجذر فاننا ندعو تلك الشجرة الشجرة ذات الجذر (*rooted tree*) .انظر الشكل(2-8) ادناه الذي يبدو انه مقلوب الشجرة الطبيعية , اي v1 تمثل الجذر وبقية النقاط تمثل الاغصان. ان الجذر يعتبر مستوي الصفر , اما مستوي اي نقطة v هي طول المسار البسيط من الجذر اليها, وبذلك فان النقطة v1 تكون في المستوي الصفر , و v2 و v3 كلاهما في المستوي **1** , اما النقاط v4 , v5, v6 , و v7 فهي في

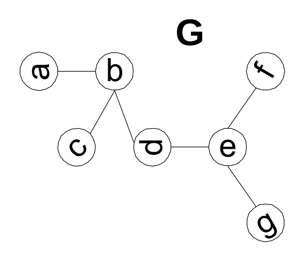
المستوي **2** , واما ارتفاع الشجرة فهو العدد الاكبر للمستويات التي تبنى منها الشجرة , اي في الشكل ادناه ارتفاعها يساوي 2.



الشكل (8-2) يوضح الشجرة ذات الجذر (*rooted tree*)

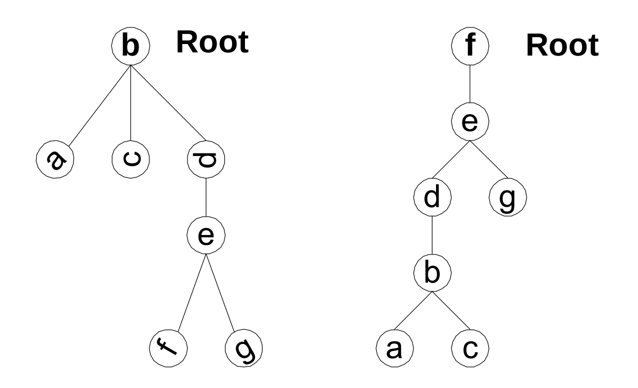
مثال 8-1:

ارسم اشجار الجذور التي تنتج من اختيار النقطة **b** اولاً ومن ثم النقطة **f** كجذور في الشجرة المبينة بالشكل (8-3) ادناه:



شكل (8-3) يمثل مخطط الشجرة في المثال (8-1)

الحل: ستكون الاشجار ذات الجذور b و f كما في الشكل (4-8) :

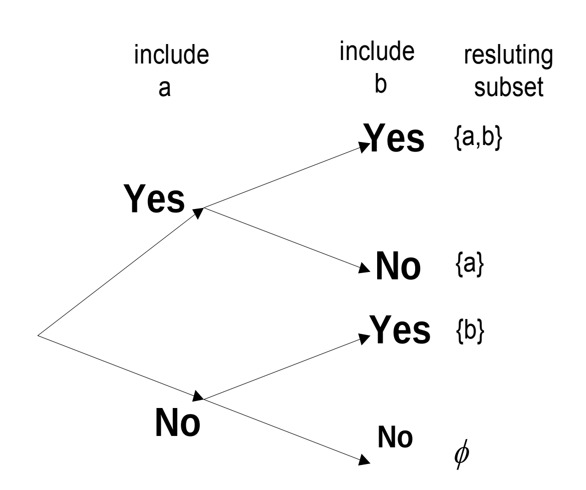


شكل (4-8) يمثل مخطط الاشجار ذوات الجذور في **b** و **f** للشكل (3-8)

ان الاطراف (*nodes*) "**a** , **c** , **f** , و **g** " الجزء الايسر مـن الشـكل (4-8) , تمثـل اطـراف خارجيـة (*external nodes*) للشجرة لعدم وجود فروع لها (*children*) , بينما **b** , **d** , و **e** تمثل اطراف داخلية (*internal nodes*) لوجود فـروع لهـا , وهذا ينطبق على كل الاشجار.

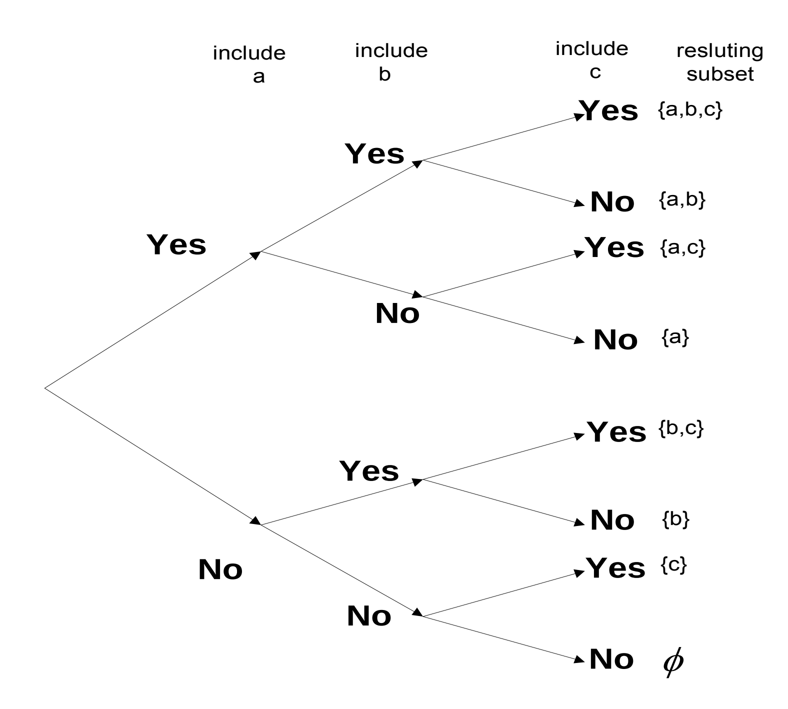
ويمكننا تمثيل امور اخرى باستخدام مخطط الشجرة (*tree diagram*), فمثلاً ناقشـنا في الفصـل الثـاني المجموعـات الجزئيـة لمجموعة معطاة, وهنا يمكننا تمثيل ذلك من خلال مخطط الشجرة (*tree diagram*), حيث نتصـور ان عنـاصر المجموعـة قد وضعت في قائمة ومن ثم نبحث في كل عنصر ونقرر فيما لو كان ذلك العنصر ينتمي الى المجموعة الجزئية .

على سبيل المثـال تصـوَّر ان لـدينا المجموعـة $\{a,b\}$. اذن مخططنـا يشـتمل عـلى خطـوتين كـما هـو في الشـكل (5-8) ادناه:ان الشكل يوضح بشكل جيد ان عـدد المجموعـات الجزئيـة الكُليِّـة (*resulting subsets*) هـي $2x2$ او 4. وهنـا اذا كانت المجموعة تشتمل على العنصر **a** (اي Yes) , والعنصر **b** (ايضاً Yes) , فـان المجموعـة الجزئيـة الناتجـة *resulting* (*subset*) هي $\{a,b\}$, اما اذا كانت المجموعة تشـتمل عـلى العنصرـ **a** , لكنهـا لاتشـتمل عـلى **b** (اي No) , اذن المجموعة الجزئية هي $\{a\}$. وهكذا تكون المجموعات الجزئية **4** .



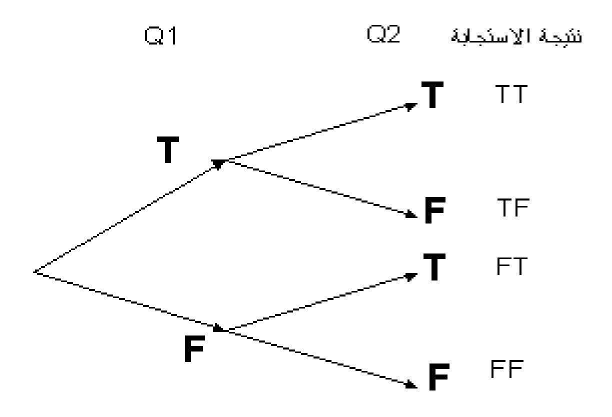
الشكل (5-8) يوضح مخطط الشجرة لبيان عدد المجموعات الجزئية للمجموعة $\{ a,b \}$

وبنفس الطريقة اعلاه , فان المخطط التالي هو لتمثيل المجموعات الجزئية لمجموعة مكونة من 3 عناصر $\{ a,b,c \}$,والذي يُبيِّن ان تلك المجموعة يكون لها $2x2x2$ اي 8 من المجموعات الجزئية . انظر الشكل (6-8) ادناه.



الشكل (6-8) يوضح مخطط الشجرة لبيان عدد المجموعات الجزئية للمجموعة $\{ a,b,c \}$

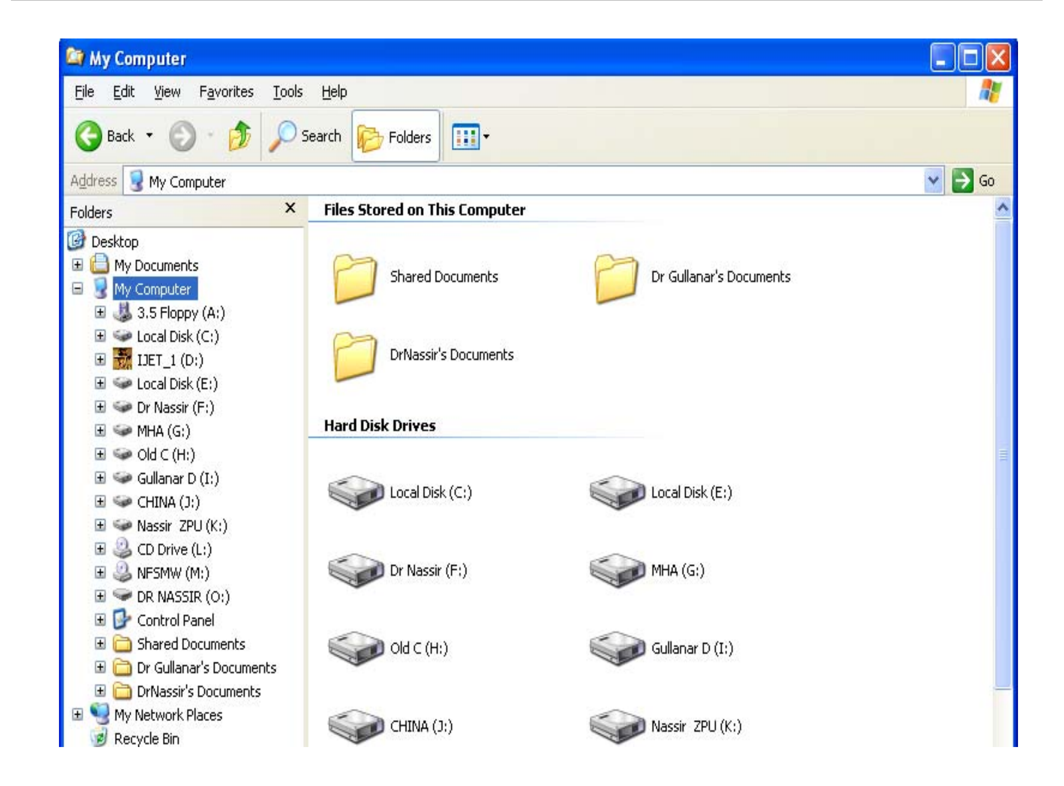
وهكذا لمجموعة مكونة من n من العناصر سيكون مخطط شجرتها مكون من $2x2x2.......x2$ اي 2^n من المجوعات الجزئية. ويمكننا ان نمثل الاجابة على سؤالين بالمخطط التالي حيث نكون لدينا 4 طرق مختلفة للاجابة , انظر الشكل (8-7) :



الشكل (8-7) يبين مخطط الشجرة لاحتماليات الاجابة على سؤالين

واذا اردنا شراء سيارتين مختلفتين ولكل سيارة لدينا ثلاثة الوان من الخيارات فسيكون مخطط الشجرة يحتوي 6 خيارات .

ومرة اخرى اذا انتقلنا الى علم الحاسوب, فنجد ان علم الحاسوب يستخدم مخطط الشجرة بكثرة لانها مهمة في تمييز وربط البيانات *data* في قاعدة البيانات (*data base*), كما ان الاشجار (*trees*) تمثل الاستعمال الواسع لفئات المخططات (*subclasses graph*) , حيث تعتبر الشجرة نوع خاص من المخطط *graph* والتي تظهر في العديد من تطبيقات نظرية المخططات (*graph theory*) .ومن انواع الاشجار(*subclasses of trees*) على سبيل المثال :الاشجار ذوات الجذر ***rooted*** (*trees*)واشجار التفرُّع الثنائي (binary trees . ومثال اخر على الشجرة في الحاسوب هو لو استخدمنا مستكشف النوافذ (*windows explorer*) لمعرفة مكونات الحاسوب , فاننا سنرى شجرة الحاسوب على اليسار من النافذة ومكونات كل جزء مختار من هذه الشجرة سيكون في النصف الثاني (اليمين) من النافذة , انظر الشكل (8-8) التالي.



شكل(8-8) الذي يبين شجرة الحاسوب باستخدام مستكشف النوافذ windows explorer

ويمكننا ايضا اسـتخدام الشـجرة لتمثيـل عمليـة بنـاء العمليـة المشـفرة في انظمـة التشـفير مثـل نظام هـوفمان للتشـفير (**Huffman code**). كما ان للاشجار تطبيقات عديدة , عـلى سـبيل المثـال منهـا مـايعرف بالاشـجار الفرعيـة *spanning trees*) ,واشجار القرار (*decision trees*) , واشجار اللعبة (*game trees*).

2-8 انشاء شجرة تشفير هوفمان Huffman Tree Creation

1-2-8 تشفير هوفمان Huffman Coding

ان الطريقة الشائعة لتمثيل الحروف والرموز والارقام (*characters*) داخل الحاسوب تتم باستخدام طول ثابت من البتتات (*fixed-length bits*) , على سبيل المثال استخدام نظام اسكي .لكن باستخدام تشفير هـوفمان فـان تلـك الحـروف والرمـوز والارقام يتم تمثيلها بطول مختلف من البتتات (*variable length bit strings*) , حيث الرمـوز الكثـيرة الاسـتخدام تمثَّـل بـ ببتتات قليلة وعكسها صحيح ايضاً.

من جانب اخر ان التكنيك الشائع لازالة تشفير البيانات الفائضة هو تشفير هوفمان (**Huffman coding**), فعند تشـفير رموز معلومات المصدر (*coding the symbols of an information source*), فان طريقة تشفير هوفمان تنتج اقـل عـدد ممكن لرموز التشفير لكل رمز في المصدر الاصلي للمعلومات .ان الخطوة الاولى في طريقـة هـوفمان تتمثـل بتخفـيض رمـوز المصدر الاصلي (*source reduction*) وذلك من خلال ترتيب احتماليات الرموز تحت الاهـتمام مـن الاعـلى الى الادنى , ثـم جمع رمزي الاحتمالية الاوطأ للحصول على احتمالية واحدة التي تحل محلهم في خطوة التقليل التالية , انظـر الشـكل (8-9) التالي الذي يمثل معلومات المصدر الاصلي (*original source*) الممثلة بستة رموز مع احتمالياتهـا التـي رتبـت مـن 0.4 الى 0.04 , وكذلك خطوات تقليل معلومات المصدر (*source reduction*) من 6 الى 2 (الممثلة بـ 0.6 و 0.4)

Original source		Source reduction			
Symbol	Probability	1	2	3	4
a_2	0.4	0.4	0.4	0.4	0.6
a_6	0.3	0.3	0.3	0.3	0.4
a_1	0.1	0.1	0.2	0.3	
a_4	0.1	0.1	0.1		
a_3	0.06	0.1			
a_5	0.04				

شكل(8-9) خطوات تقليل معلومات المصدر (*source reduction*) بطريقة تشفير هوفمان

وللحصول على اول تقليل فان الاحتماليتين في الاسفل (0.04 و 0.06) يتم جمعهـما للحصـول عـلى رمـز مركّـب باحتمالية 0.1 , ويتم مرة ثانية ترتيب الاحتماليا من الاعى الى الادنى ان تطلب ذلك . ثم يستمر جمع اقل احتماليتين وهنـا نجمع 0.1 مع 0.1 لنحصل على 0.2 , وبمـا انهـا اكـبر مـن 0.1 الاخـرى , اذن تصـعد 0.2 الى الاعـلى وتنـزل 0.1 اسـفلها ونستمر الى ان نصل الى التخفيض الرابع لنحصل على 0.6 و 0.4 , حيث يكون مجموعهما 1 , اي 100% .

وهنا تبدأ المرحلة الثانية من طريقة هوفمان وهي تشفير كل مصدر تم تقليله مبتدأين بالرمز 0 للاحتمالية الاكبر (0.6) و1 للاحتمالية الاقل او العكس من اجل تمييزهما, ونرجع باتجاه عكسي بالقيم صفر او واحد , فلو اعطينا صفراً لكل الرموز التي كونّت 0.6 , اي 0.3 و 0.3 ومن ثم اضافة 0 للاول و 1 للثاني من اجل تمييزهما مرة ثانية , اي تصبح 0.3

لها الرمز 00 , وهي مميزة عن 0.3 الثانية التي ستاخذ الرمز 01, بينما 0.4 تاخذ 1 لاننا اعطينا 0 الى 0.6 في البداية . ثم نستمر بالاتجاه العكسي ونعطي مكونات 0.3 التي هي 0.2 و 0.1 رمز 0.3 الاصلي (اي 01) ثم نضيف 0 للاول و 1 للثاني لتصبح 0.2 تاخذ الرموز 010 , و 0.1 تاخذ الرموز 011 وهكذا نستمر وبنفس الاسلوب الى ان نصل الى كل ارقام الشفرات (*codes*) التي تمثل الرموز الاصلية (انظر عمود الشفرة (*code*) في يسار الشكل (8-10) ادناه.

Original source			Source reduction			
Sym.	Prob.	Code	1	2	3	4
a_2	0.4	1	0.4 1	0.4 1	0.4 1	0.6 0
a_6	0.3	00	0.3 00	0.3 00	0.3 00	0.4 1
a_1	0.1	011	0.1 011	0.2 010	0.3 01	
a_4	0.1	0100	0.1 0100	0.1 011		
a_3	0.06	01010	0.1 0101			
a_5	0.04	01011				

شكل(8-10) طريقة هوفمان لتشفير كل مصدر تم تقليله في شكل (8-7) مبتدأين بالرمز 0 و 1

فلو كانت لدينا صورة رقمية وكان r_k يمثل عدد مستويات التدرُّج اللوني الرمادية (*gray level*) فيها , فان احتمالية حدوث كل مستوي رمادي تتمثل بالعلاقة :

$$p_r(r_k) = \frac{n_k}{n} \qquad k = 0, 1, 2, \ldots , L - 1$$

حيث n_k يمثل عدد المرات التي يحدث فيها المستوي الرمادي المعني , و **n** تمثل عدد البكسل المكونة للصورة , و **L** يمثل عدد المستويات الرمادية , فان معدل طول هذه الشفرة (code) يمكننا الحصول عليه من العلاقة التالية :

$$L_{avg} = \sum_{k=0}^{8-1} l_2(r_k)\, p_r(r_k)$$

= **1** (0.4) + **2** (0.3) + **3** (0.1) + **4** (0.1) + **5** (0.06) + **5** (0.04) =2.2 *bits/ symbol*

لاحظ ان الارقام **1, 2, 3, 4, 5, و 5** تمثل عدد الشفرات (codes)النهائية (*digit numbers*) مقابل كل احتمالية , فمثلاً 0.4 يقابلها 1 (*one digit*) , 0.3 يقابلها 00 اي (*two digits*) , وهكذا 0.04 يقابلها 01011 اي (*5 digits*) .

اما معدل المعلومات (average information) لكل نتاج للمصدر (source output) يدعى الانتروبي (entropy) للمصدر

$$Entropy = -\sum_{j=1}^{J} p_r(r_k) \log_2 p_r(r_k)$$

Entropy =-[0.4 $\log_2(0.4)$ + 0.3 $\log_2(0.3)$ +0.1 $\log_2(0.1)$ + 0.1 $\log_2(0.1)$ + 0.06

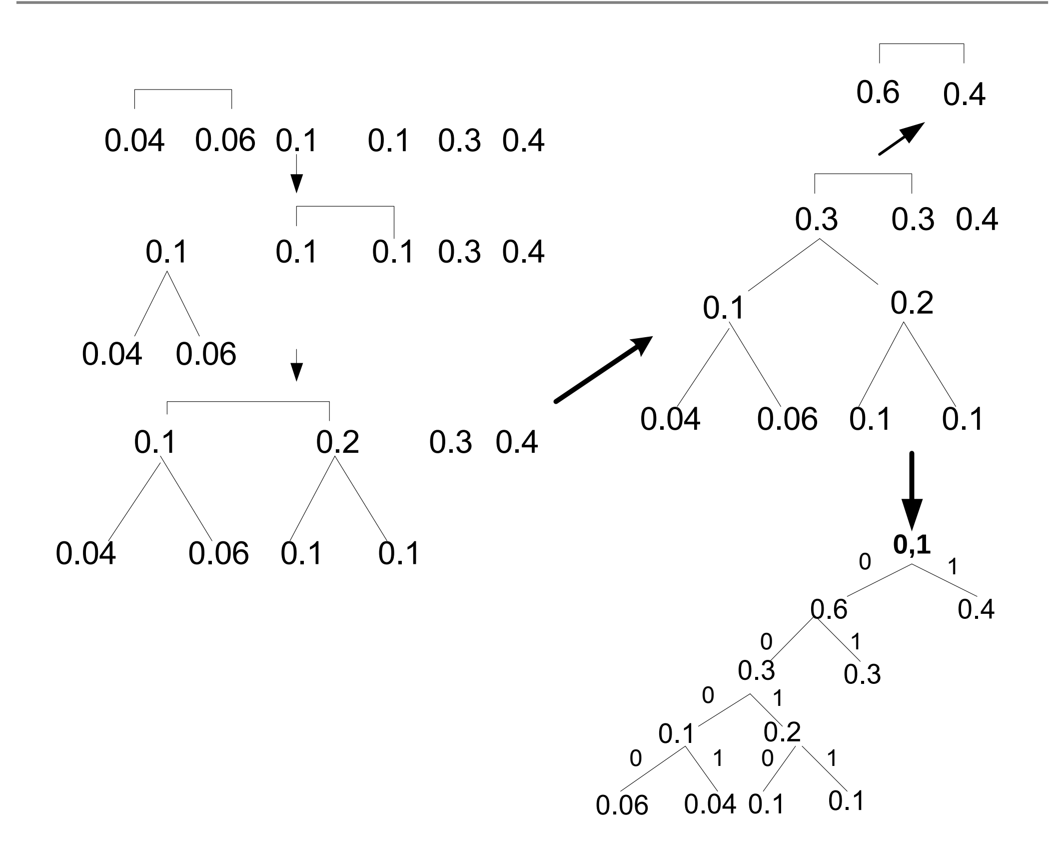
$\log_2(0.06)$ + 0.04 $\log_2(0.04)$] =2.14 bit / symbol

حيث ان

- **$\log_2(x)$** $=\log_{10}(x) / \log_{10}(2)$

لقد تم اعلاه استخدام لوغاريتم للاساس 2 من اجل جعل الانتروبي بوحدات البتتات لكل رمز , وان نتيجة كفاءة او فعالية تشفير هوفمان هي : ***The resulting Huffman code efficiency is = 2.14 / 2.2 = 0.973***

ان طريقة هوفمان اعلاه يمكننا تمثيلها بواسطة الشجرة (Tree) خطوة بخطوة انظر الشكل (8-11) كما يلي:



شكل(8-11) شجرة هوفمان التي انشأت لـ 6 رموز من البيانات

مثال 8-2:

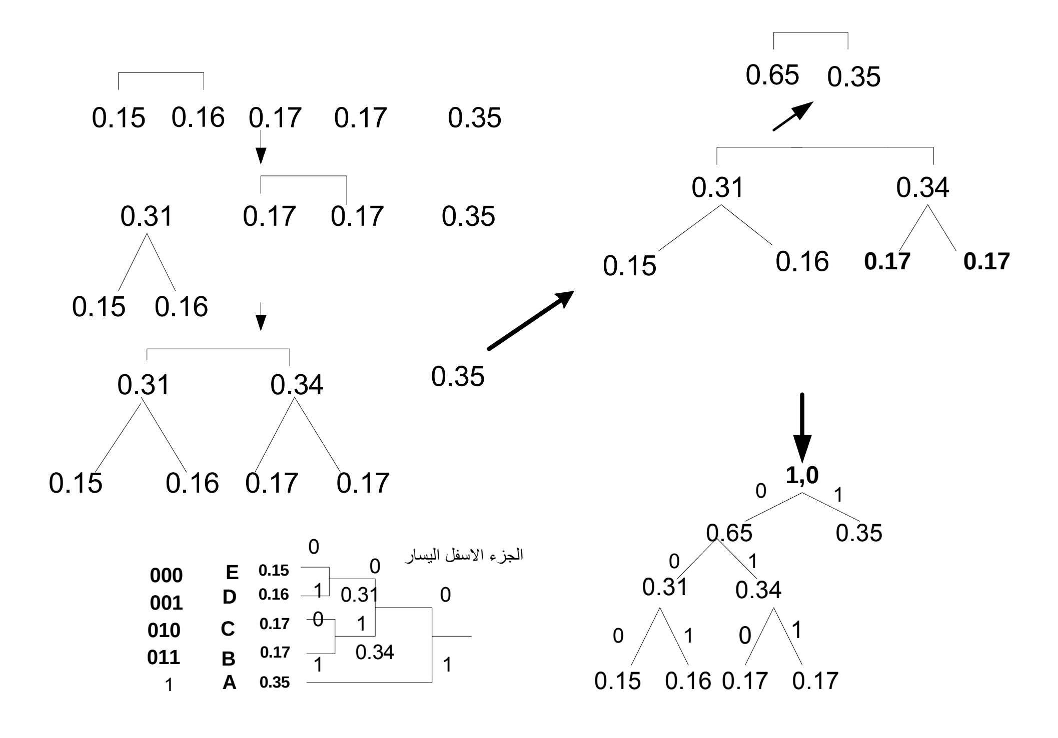
انشئ شجرة هوفمان للحروف الخمسة التالية A , B , C , D , E بالاحتماليات 0.35 ,0.17 ,0.17 ,0.16 ,0.15 على التوالي.

بنفس الاسلوب الذي اعتمدناه في تشفير هوفمان فاننا نرتب الاحتماليات من الاعلى الى الادنى ثم نجمع اصغر احتماليتين للحصول على احتمالية مركبة , فمثلاً نجمع 0.15 مع 0.16 لنحصل على 0.31 لتصبح اسفل 0.35 في الترتيب التالي , انظر العمود 5 , وهكذا نجمع 0.17 مع 0.17 لنحصل على 0.34 التي تصبح فوق 0.31 في العمود 7 , وهكذا نجمع 0.34 مع 0.31 لنحصل على 0.65 التي ستكون في قمة العمود 8 لنعطيها صفرا ونعطي 0.35 واحد (1) , لنرجع بالعكس الى مكوناتهما الى ان نصل الى الشفرات (codes) النهائية في العمود 2 , انظر جدول (8-1).

1	2	3	4	5	6	7	8	9
A	**1**	0.35	**1**	0.35	**1**	0.35	0.65	**0**
B	**011**	0.17	**00**	0.31	**01**	0.34	0.35	**1**
C	**010**	0.17	**010**	0.17	**00**	0.31		
D	**001**	0.16	**011**	0.17				
E	**000**	0.15						

جـــــــدول (8-1). يمثـــــــل خطـــــــوات تشـــــــفير A , B , C , D , E بالاحتماليـــــــات 0.35 ,0.17 ,0.17 ,0.16 ,0.15 على التوالي.

ويمكن تمثيل ذلك بالشجرة (Tree) خطوة بخطوة كما هوفي الشكل (8-12) ادناه :



شكل(8-12) شجرة هوفمان للحروف A , B , C , D , E بالاحتماليات 0.35 ,0.17 ,0.17 ,0.16 ,0.15 على التوالي.

لاحظ انه ببساطة يمكننا معرفة الشفرات (codes) لكل حرف وذلك من خلال الجزء الاسفل من اليسار وبالقراءة للشفرات 000 او 001 وذلك من اليمين باتجاه الشمال.

اذن معدل طول (average length) الشجرة في الشكل (8-12) هو:

$$L_{Huff} = .35.1 + .17.3 + .17.3 + .16.3 + .15.3 = 2.3 \ \textit{bits/ symbol}$$

لاحظ مرة ثانية ان الارقام **1, 3** المضروبة بالاحتماليات تمثل عدد (الشفرات) الكود النهائي (*digit numbers*) مقابل كل احتمالية ,وهي جميعها 3 ارقام ماعدا A فهي تقابل 1 , انظر العمود 2 في الجدول (8-1)

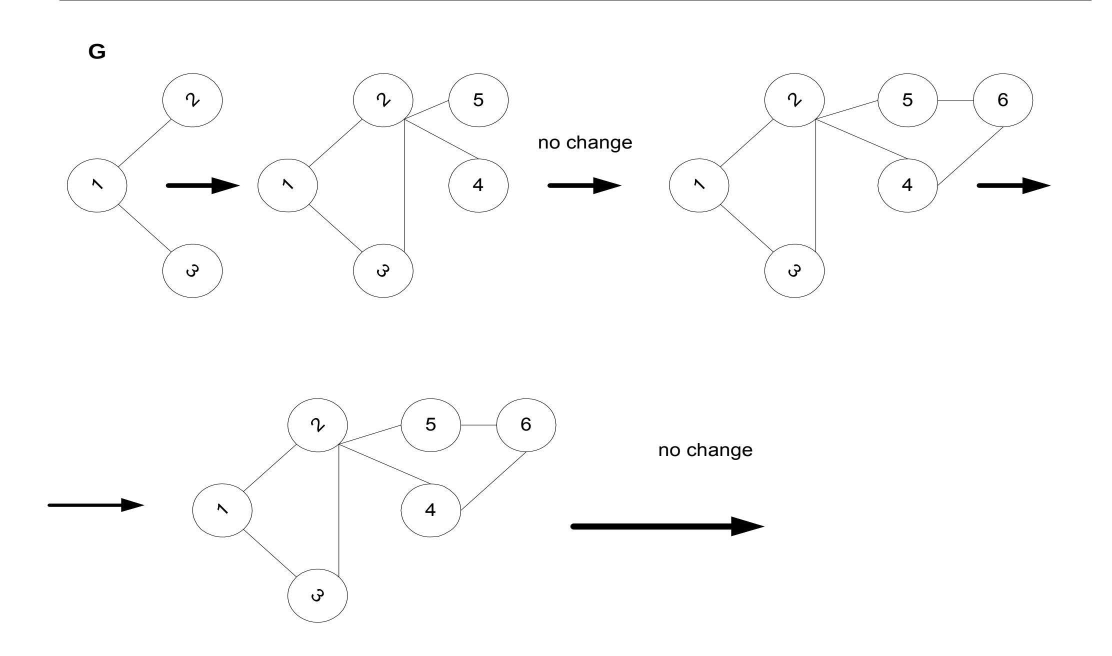
مثال 8-3:

ارسم المخططات الموصوفة بطريقة تجاور القوائم (*adjacency lists*) التالية وبيِّن ايهما يمثل شجرة(tree) .[2]

$G: \quad 1:2,3; \quad 2:1,3,4,5; \quad 3:2; \quad 4:2,6; \quad 5:2,6; \quad 6:4,5.$

$H: \quad 1:3; \quad 2:3,4,5; \quad 3:2; \quad 4:2; \quad 5:2,6; \quad 6:5.$

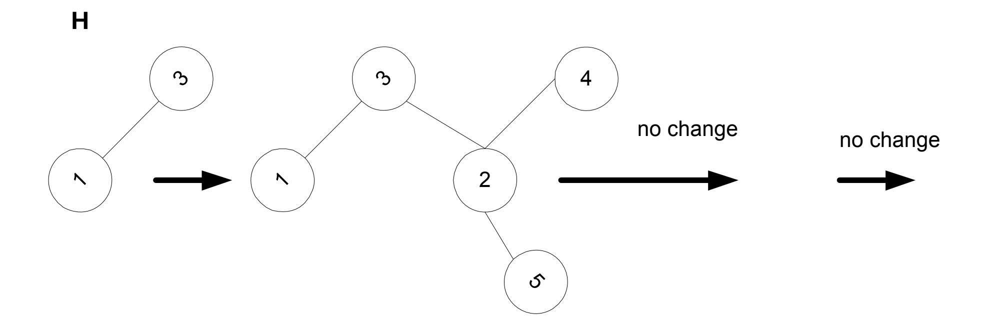
اولاً نرسم مكونات المخطط **G** خطوة بخطوة انظر الشكل (8-13) كمايلي:

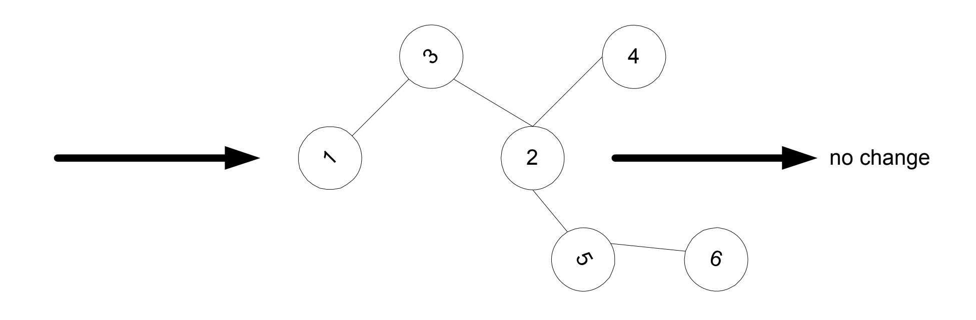


الشكل (8-13) يبين رسم مكونات المخطط G خطوة بخطوة

نلاحظ من المخطط اننا بدأنا اولاً برسم النقطة 1 التي تتصل بالنقطة 2 , و3 . ثم ننتقل الى النقطة 2 التي تتصل بـالنقاط 1, 3, 4, 5, . اذن نضيف الى الرسم السابق فقط النقطة 4 والنقطة 5 ونوصلهما بـ 2 , ونوصل 2 ايضاً بـ 3 الموجـودة اصـلاً في الرسم السابق . وبعد ذلك ننتقل الى النقطة 3 من مكونات المخطط G اعلاه لنجد انها متصلة بـ 2 , اذن نكتب لاتغيير *no change*)) , وهكذا نستمر لنصل الى المخطط النهائي . ان هذه الطريقة تسمى طريقة القوائم المتجاورة (*adjacency lists*).

وبنفس الطريقة نرسم المخطط H , انظر الشكل (8-14) وكمايلي:





الشكل (8-14) يبين رسم مكونات المخطط H خطوة بخطوة

ومن خلال المخططين نرى انهما مخططان متصلان (*connected graphs*) , ولكن نـرى ان المخطـط G لـه دورة (*Cycle*) , اما H فليس له دورة . اذن H يمثل شجرة و G لايمثلها. وبناءاً على ذلك يعرّف المخطط البسيط (simple graph) بانـه شجرة ايضا وان لكل زوج من النقاط (x و y) فيه يوجد مسار وحيد مـن x الى y . وانـه مخطـط متصـل ولكـن لـيس لـه دورة (*Cycle*). وهذه تمثل مواصفات اخرى عن الاشجار .

3-8 الشجرة الفرعية Spanning Tree

ان الشجرة لمخطـط مـا G , تـدعى شـجرة فرعيـة (*spanning tree*),اذا كانـت جـزء مـن ذلـك المخطـط (*subgraph*) وتحوي جميع النقاط (*vertices*) فيه . او بتعبير اخر لو اعطينا مخططاً G لمجموعـة V مـن النقـاط (*set of vertices*) , ومجموعة E من الحدود (*the set of edges*), فاننا نعرّف الشجرة الفرعية لـ G بانها الشجرة التـي لهـا نفـس مجموعـة النقاط V , بينما مجموعة الحدود لها هي جزء من المجموعة E .

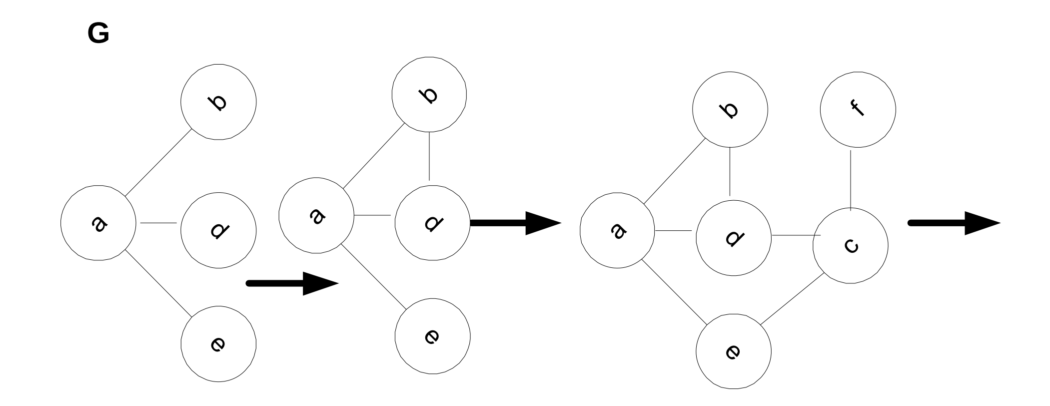
مثال 8-4:

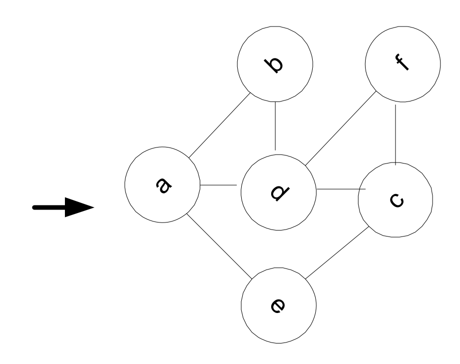
ارسم المخطط الموصوف بطريقة تجاور القوائم (*adjacency lists*) التالية ورسم الشجرة الفرعية (spanning tree) له.

$a : b, d, e; \quad b : a, d; \quad c : d, e, f; \quad d : a, b, c, f; \quad e : a, c; \quad f : c, d.$

الحل:

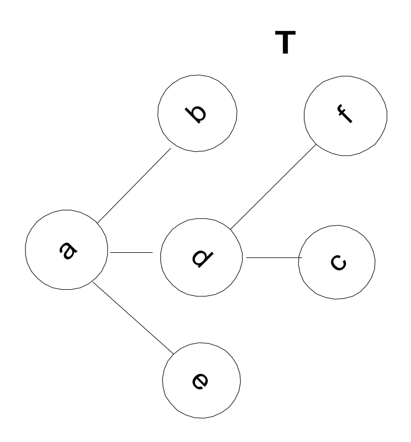
بنفس الاسلوب الذي اعتمدناه في المثال (8-3) سيكون المخطط المعني ونرمز له بـ G انظر الشكل(8-15):





الشكل (8-15) يبين رسم مكونات المخطط في المثال (8-4) خطوة بخطوة

اما مخطط الشجرة الفرعية (*Spanning tree*) ونرمز له بـ **T** فهو كما في الشكل (8-16)



الشكل (8-16) مخطط الشجرة الفرعية (*Spanning tree*) للمخطط في الشكل (8-15)

ومن خلال مخطط الشجرة الفرعية (*Spanning tree*) اعلاه نجد ان عدد الحدود (*edges*) فيها هو 5 وهو يساوي عدد النقاط (*vertices*) التي هي 6 هنا مطروحاً منها 1. وبشكل عام فان شجرة لها ***n*** من النقاط (*vertices*) يكون لها **n-1** من الحدود (*edges*).

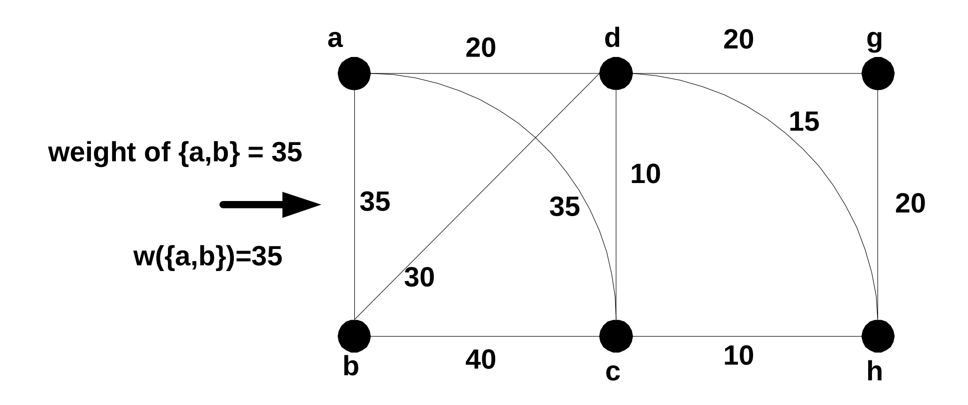
8-5 المخطط الموزون Weighted Graph

ويعني انه لدينا مخطط ولدينا عدد(وزن او قيمة) يصاحب كل حد (*edge*) فيه. اي ان المخطط الذي له عدد $w(e)$ مصاحب لكل حد e فيه و يسمى (*weighted graph*) , انظر الشكل (8-17). ان $w(e)$ تسمى وزن الحد **e** . ان الشكل التالي يمثل الطريقة القياسية لرسم المخطط الموزون. وفي الاشجار الفرعية (*spanning tree*) ايضا لدينا عدد يصاحب كل شجرة يسمى مجموع اوزان الحدود (*the sum of the weights of its edges*) .

وفي بعض المسائل نود ان يكون ذلك المجموع اقل مايمكن وفي بعضها نود ان يكون اكبر مايمكن . ان اقل مجموع كلي (*minimum total*) للشجرة الفرعية الموزونة (*weight spanning tree*) , او مايسمى (*minimum spanning tree*) لمخطط G موزون هو الشجرة الفرعية (*spanning tree*) لـ **G** بخاصية ان مجموع الاوزان لحدودها (*its edges*) هو ليس اكبر من مجموع اوزان الحدود لاية شجرة فرعية اخرى لـ G . ان مجموع اوزان الحدود الكلي لشجرة يسمى الوزن الكلي للشجرة. فمثلاً في الشكل (8- 17) مجموعة الحد (*edge set*) التالية:
$\{\{a,b\},\{a,c\},\{a,d\},\{d,g\},\{d,h\}\}$ هي مجموعة الحد للشجرة الفرعية , والوزن الكلي لها هو 125 وكمايلي:

$$w(\{a,b\}) = 35,\quad w(\{a,c\}) = 35,\quad w(\{a,d\}) = 20,\quad w(\{d,g\}) = 20, \text{and}\quad w(\{d,h\}) = 15$$
$$\therefore \text{the total weight} = 125.$$

وهنا يمكننا ان نحذف w{a,c} , ونبدِّله بـ {d,c} , وهذا يقلل الوزن الى 100 . اذن هذه الشجرة الفرعية ليست *minimum weight spanning tree*).



شكل(8-17) المخطط الموزون ***weighted graph***

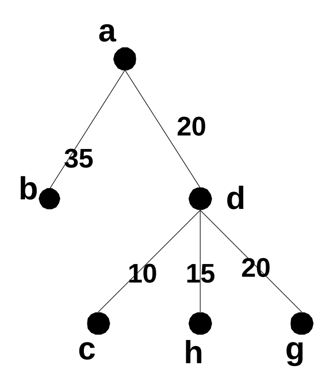
مثال 8-5:

اوجد مجموعة الحد (*edge set*) لمسار الحد الادنى للشجرة الفرعية المتمركزة في النقطة **a** للشكل (8-17), وكذلك اوجد مسار وزن الحد الادنى (*a path of minimum weight*) من **a** الى **h** في المخطط المبيَّن في الشكل (8-17) ايضاً.

ان مجموعة الحد (*edge set*) لمسار الحد الادنى للشجرة الفرعية المتمركزة في النقطة **a** , هي:

$$\{\{a,b\},\{d,c\},\{a,d\},\{d,g\},\{d,h\}\}$$

ان الشجرة التي تمثل ذلك والتي جذرها في **a** (*rooted tree*) يوضحها المخطط المبين بالشكل (8-18)



الشكل (18-8) مجموعة الحد (*edge set*) لمسار الحد الادنى للشجرة الفرعية المتمركزة في النقطة **a**

ونرى من الشكل ان المسار من **h** رجوعاً الى الجذر **a** هو:

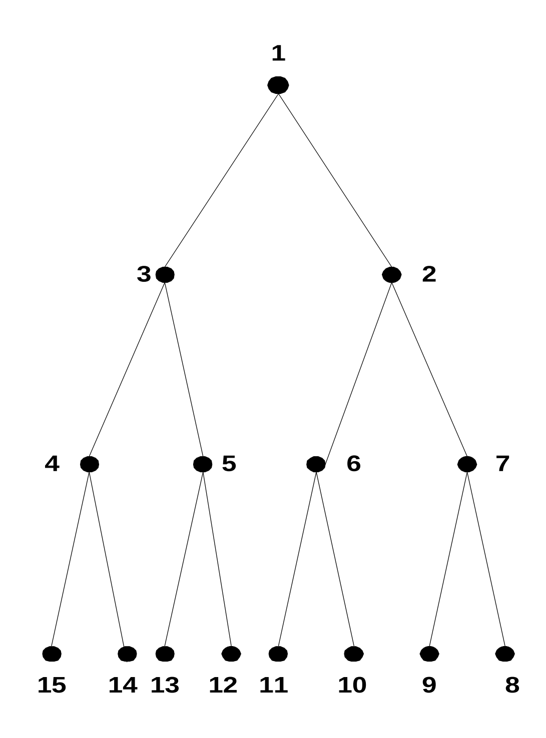
$$h\{d,h\}d\{a,d\}a$$

وان طول المسار هو:

$$w(d,h)+w(a,d)=15+20=35$$ ▲

6-8 الشجرة ثنائية التفرع Binary Tree

من جانب اخر ان الشكل (19-8) يبين شجرة ثنائية التفرع كاملة (*full binary tree*) ذات 3 مستويات , والتي تعني انه يخرج من كل نقطة فرعان كما مر بنا في اشجار التشفير (انظر الجزء (2-8) في هذا الفصل.

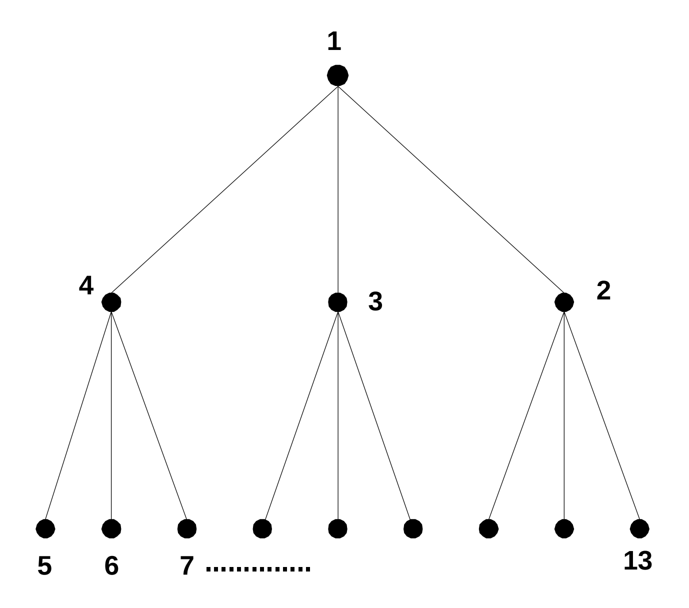


الشكل (8-19) يبين شجرة ثنائية التفرع كاملة (*full binary tree*)

ذات 3 مستويات

7-8 الشجرة ثلاثية التفرع Ternary Tree

وتعني انه يخرج من كل نقطة ثلاثة فروع , لاحظ النقاط **1, 2, 3, و 4.** ان الشكل (8-20) يبين شجرة ثلاثية التفرع كاملة (*full ternary tree*) ذات مستويين.



الشكل (8-20) يبين شجرة ثلاثية التفرع كاملة (*full ternary tree*) ذات مستويين

تمارين الفصل الثامن

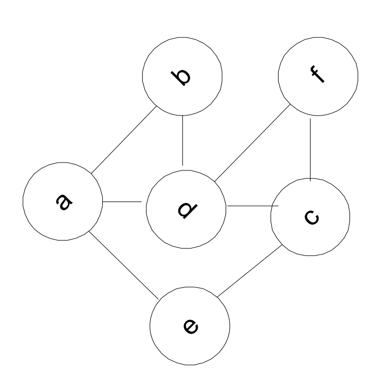
1-Create Huffman tree of the following 6 symbols with the probability values $0.15, 0.16, 0.17, 0.17, 0.35$ **respectively .**

2- Draw the graphs described by the adjacency lists below, explain which a tree is?

G : 1 : 2,3; 2 : 1,3,4,5; 3 : 2; 4 : 2,6; 5 : 2,6; 6 : 4,5.

H : 1 : 3; 2 : 3,4,5; 3 : 2; 4 : 2; 5 : 2,6; 6 : 5.

3-If you have given the following graph, what is the spanning tree? Explain.



References **المراجع**

1- Johnsbaugh Richard ," Discrete Mathematics". Prentice Hall,5th Ed. 2001.

2- P. Bogart, "Discrete Mathematic",. D.C Health and company, 1988.

3- Jack R. Britton and Ignacio Bello "Topics In Contemporary Mathematics". Dellen publishing companay, 1989.

4- Ross A. Kennth, "Discrete Mathematics ", Prentice Hall, 1992.

5- Donald F. Stanat, Discrete Mathematics In Computer Science. Prentice Hall, 1977.

6- Tremblay, "Discrete Mathematic Structure With Applications To Computer Science.

7- Larry J. Goldstien, David C. Lay, and Daid I. Schneider "Calculus And Its Applications ",. Prentice Hall, 4th Ed., 1987.

8- Deitel and Deitel , "C++: How to Program", Prentice Hall, 4th Edition, 2003